UNIVERSITAT POLITÈCNICA DE VALÈNCIA

DSIC
DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

Diseño e implementación de una herramienta de soporte para el modelado de procesos de negocio con Análisis Comunicacional

Trabajo Fin de Máster

Máster Universitario en Ingeniería y Tecnología de Sistemas Software

Departamento de Sistemas Informáticos y Computación

Alumno: Luis Eduardo Flores Mendez
Tutor: Óscar Pastor López

2020-2021

RESUMEN

El Análisis Comunicacional (AC) es un método de modelado para procesos de negocio, que permite la especificación de requisitos para sistemas de información organizacionales. Hoy en día, el método cuenta con una herramienta para soportar el método, basada en Eclipse Modelling Framework (EMF). Sin embargo, tanto las limitaciones propias de EMF como la necesidad de integración de los modelos de AC con otros modelos, en diversas plataformas tecnológicas plantea el desafío de diseñar e implementar una nueva herramienta.

En este trabajo se presenta el diseño e implementación de una herramienta de modelado de AC basada en web, utilizando la metodología de Ciencia de Diseño. La herramienta permite cubrir las distintas especificaciones de AC: realizar el diagrama de eventos comunicativos, asociar a cada evento comunicativo una estructura de mensaje. La contribución de este trabajo es resolver un problema de investigación en ingeniería relativo a diseñar la mejor forma de soportar los atributos de calidad y funcionalidad, considerando el estado del arte y práctica en herramientas de modelado, e implementar dicho diseño.
La herramienta desarrollada permite además dar soporte a la investigación de la comprensibilidad del método de AC y de un método holístico de producción de software.

Palabras Clave: Análisis Comunicacional, Diagramas de evento comunicativo, Estructura de mensajes.

ABSTRACT

Communication Analysis (CA) is a modeling method for business processes, which allows the specification of requirements for organizational information systems. Nowadays, the method has a tool to support the method, based on Eclipse Modelling Framework (EMF). However, both the limitations of EMF and the need to integrate CA models with other models in different technological platforms pose the challenge of designing and implementing a new tool.

This paper presents the design and implementation of a web-based CA modeling tool, using the Design Science methodology. The tool allows to cover the different specifications of CA: to perform the diagram of communicative events, to associate to each communicative event a message structure. The contribution of this work is to solve an engineering research problem related to design the best way to support quality and functionality attributes, considering the state of the art and practice in modeling tools, and to implement such design. The developed tool also allows to support the investigation of the comprehensibility of the CA method and of a holistic method of software production.

Keywords: Communication Analysis, Communication Event Diagrams, Message Structure.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

Capítulo 1

Motivación

En la ingeniería de software, el desarrollo guiado por modelos, sitúa en el centro de la actividad a los modelos, dejando la implementación sólo como un requerimiento técnico. Los modelos se convierten en una herramienta de alto nivel, la cual permite a agentes externos al área de software comprender la solución. Dado esto se han desarrollado variadas técnicas de modelado que atacan el problema a distintos niveles de abstracción, entre ellas, modelado organizacional, modelado de procesos de negocios con análisis comunicacional y modelado de sistemas. Cada técnica de modelado puede interoperar con otras a través de transformaciones mediante herramientas de apoyo; sin embargo, estas herramientas no se comunican entre sí, y no son capaces de cumplir el principal objetivo del desarrollo guiado por modelos que es la autogeneración del código. En el presente trabajo me centrare en el modelado de procesos de negocio con análisis comunicacional.

El Análisis Comunicacional (AC) es un método de ingeniería de requisitos que propone describir los procesos de negocio desde una perspectiva comunicativa; analizando las interacciones comunicativas entre el sistema de información y su entorno [1] y especificando los procesos de negocio mediante i) diagramas de eventos comunicativos, una técnica de modelado gráfico de notación semejante a los diagramas de actividad, y ii) plantillas de especificación de eventos comunicativos que es una técnica de modelado destinada a especificar los requisitos asociados a un determinado evento comunicativo y la estructura de mensajes compuesto por subestructura de datos.

En el artículo A Models-to-Program Information Systems Engineering Method [2], se plantea demostrar que es posible diseñar un método de producción de software holístico para generar código a partir de diferentes niveles de abstracción (desde el espacio del problema hasta el espacio de la solución). La propuesta plantea que es posible producir software usando modelos de requisitos a código, partiendo por el modelado organizacional, pasando por el de procesos de negocio, y llegando hasta modelos de sistema que generan código automáticamente. Esto implica la integración conceptual de métodos de modelado y técnica de sus herramientas de soporte.

Hoy en día, el modelado de procesos de negocios a través de Análisis Comunicacional cuenta con una herramienta para soportar el método, basada en Eclipse Modelling Framework (EMF). Sin embargo, tanto las limitaciones propias de EMF como la necesidad de integración de los modelos de AC con otros modelos, plantea el desafío de diseñar e implementar una nueva herramienta esperando la integración de AC con otros modelos mencionados anteriormente. Para ello, haré uso de librerías como mxGraph y React junto al entorno de desarrollo Visual Studio Code para el diseño e implementación de una herramienta de soporte para el modelado de procesos de negocio con Análisis Comunicacional. Se cuenta para ello con la ayuda de la documentación que hay al alcance en internet sobre mxGraph[1] y React[2], permitiendo el aprendizaje sobre estas librerías y las diferentes funciones que dispone.

Personalmente, el desarrollo de software es la parte que me apasiona, y es lo que me motiva en el presente trabajo para el diseño e implementación de esta herramienta de soporte para el modelado de procesos de negocio con Análisis Comunicacional que permitirá la realización de diagramas de eventos comunicativos y a su vez asociar a cada evento comunicativo una estructura de mensaje esperando la integración de AC con otros modelos. De esta forma el objetivo es poder aportar al centro de investigación producción de software de la UPV (PROS) y la comunidad científica en el tema de modelamiento conceptual y desarrollo dirigido por modelos con una herramienta, que fomente, futuros trabajos.

.

[1] definición de mxGraph disponible en: **https://jgraph.github.io/mxgraph**

[2] definición de React disponible en: **https://es.reactjs.org/**

[3] PROS disponible en: **http://www.pros.webs.upv.es/**

Capítulo 2

Metodología

Para la metodología de trabajo se utilizará el método de Ciencias del Diseño (Design Science), [3], dado que se busca diseñar una herramienta para su integración dentro de un método de producción de software dirigido por modelos, teniendo en cuenta el diseño de un producto dividido en componentes como se muestra en la siguiente figura:

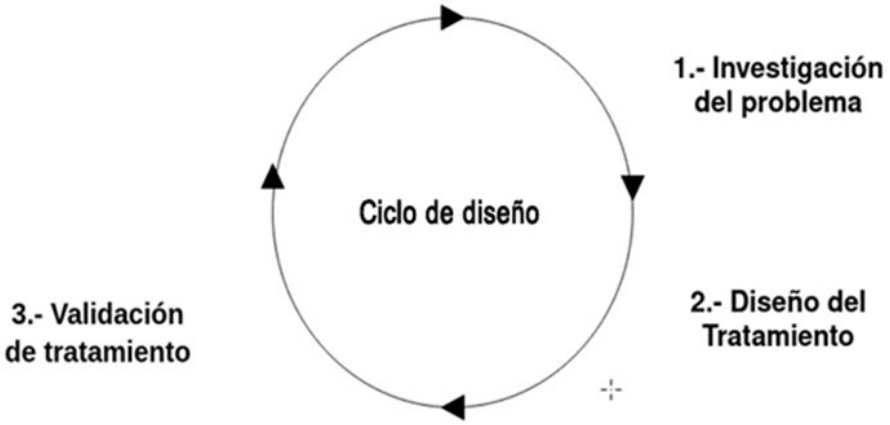

Figura 1: Diagrama de la Metodología – Ciclo de diseño

2.1. Investigación del Problema

En la investigación del problema se especificará el marco conceptual, correspondiente al análisis comunicacional (diagrama de eventos comunicativos, plantillas de especificación de eventos comunicativos y estructura de mensajes), y el método holístico de requisitos a código donde se propone un método para generar código a partir de diferentes niveles de abstracción. Además, se definirá el problema y se planteará los objetivos que se quieren lograr.

2.2. Diseño del tratamiento

En el diseño del tratamiento se definirán los requisitos de la solución indicando la solución actual (GREAT)[1], la arquitectura distribuida para la transformación de modelos y los requisitos de alto nivel, tanto funcionales que son los que especifican el manejo que requiere el sistema y las funciones que debe incluir; y los no funcionales que especifican las características del sistema. Por otro lado, se especificará la propuesta de la solución.

[1] GREAT disponible en: **http://www.pros.webs.upv.es/great/**

2.3. Validación del tratamiento

En la validación del tratamiento se indicará el diseño de la validación indicando el cuestionario que se realizó para la verificación de la funcionalidad de la herramienta desarrollada, así como el análisis de los resultados.

Capítulo 3

Investigación del Problema

3.1. Análisis Comunicacional

El análisis comunicacional es un método de ingeniería de requisitos que propone describir los procesos de negocio desde una perspectiva comunicativa. El objetivo es descubrir y describir las interacciones comunicativas entre el Sistema de Información y su entorno [4].

Se define interacción comunicativa como la interacción entre actores con el objetivo de intercambiar información. Dependiendo de la dirección preponderante de la comunicación, se distinguen dos tipos de interacción comunicativa:

- Interacción comunicativa entrante: Su objetivo fundamental es aportar al sistema nueva información significativa; es decir, alimenta la memoria del SI con datos relevantes para la organización, previamente desconocidos.

- Interacción comunicativa saliente: Su objetivo fundamental es distribuir datos conocidos a los usuarios; es decir, consulta la memoria del sistema para recuperar datos y presentarlos a los usuarios.

El AC propone especificar los procesos de negocio mediante diagrama de eventos comunicativos, una técnica de modelado gráfico de notación semejante a los diagramas de actividad, plantillas de especificación de eventos comunicativos que es una técnica de modelado destinada a especificar los requisitos asociados a un determinado evento comunicativo y la estructura de mensajes compuesto por subestructura de datos. [4]

a) **Diagrama de Eventos Comunicativos**: Una técnica de modelado de procesos de negocios que adopta una perspectiva comunicacional y facilita el desarrollo de un IS que apoyará esos procesos de negocios.

   El Diagrama de Eventos Comunicativos representa gráficamente la secuencia de interacciones entre un Actor Primario (que inicia la comunicación), un Actor de Apoyo (que es la interfaz de la organización en la comunicación), y el actor Receptor (que es notificado de los resultados del evento) [2].

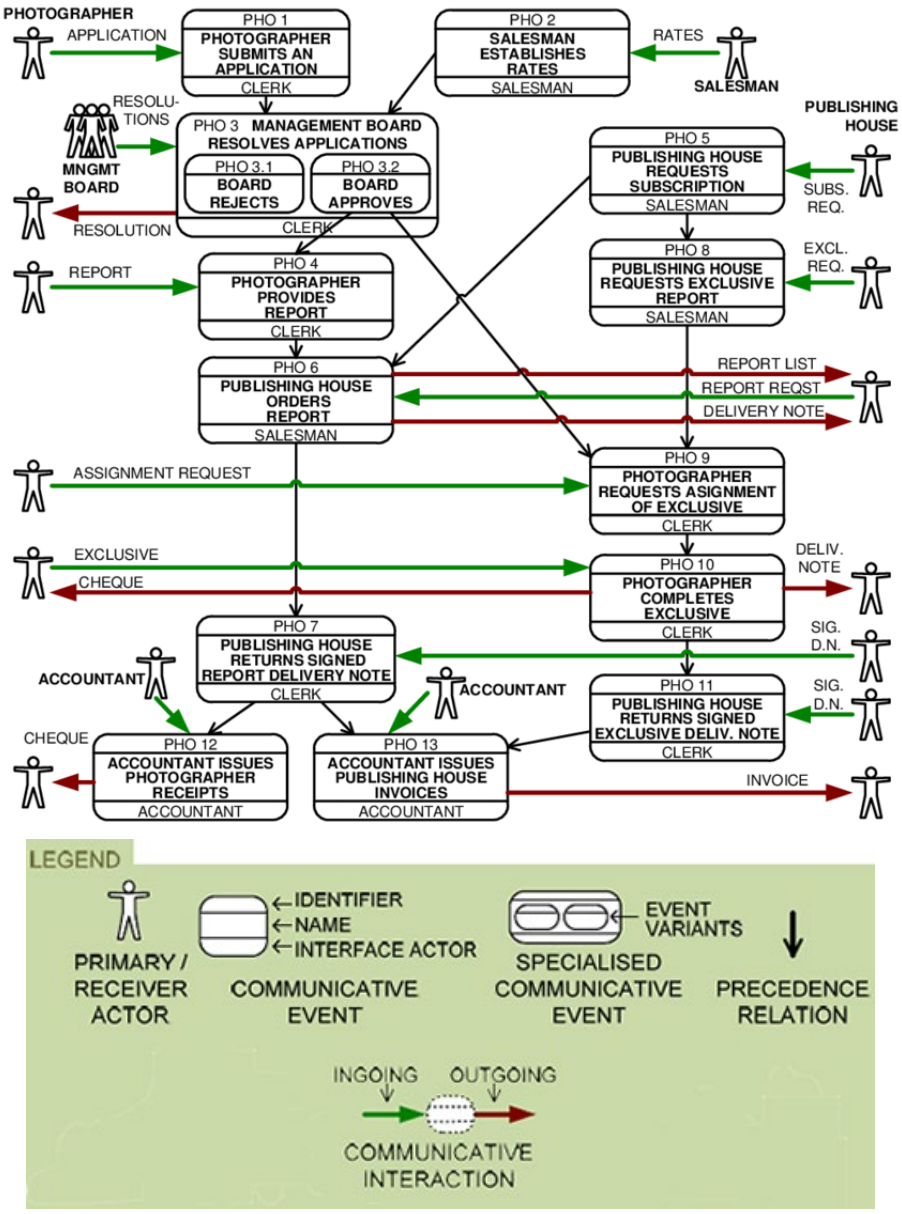

Figura 2: Ejemplo de Diagrama de evento comunicativo [5]

Leyenda:

- **Actor principal (Primary Actor)**: Es un actor organizativo que desempeña un papel primordial en el contexto de acción de un evento comunicativo. Es responsable de comunicar la nueva información significativa al sistema de información (por ejemplo, una cliente que formula una solicitud). Es habitual que el actor principal también desencadene la interacción comunicativa (es decir, inician la interacción estableciendo contacto con el sistema de organización; por ejemplo, un cliente viene y hace un pedido), pero no siempre es así (por ejemplo, un representante de ventas puede visitar a un cliente y convencer para hacer un pedido).

- **Actor receptor (Receiver actor)**: Es un actor organizativo que desempeña un papel de receptor en el contexto de acción de un evento comunicativo. Se espera que a un actor receptor se le comunique alguna información como resultado de un evento comunicativo (por ejemplo, el gerente de ventas recibe una copia del formulario de pedido).

- **Evento comunicativo (Communicative event)**: Es una transición compuesta; más específicamente, una idea unitiva que unifica varias ideas sobre una interacción comunicativa entrante y la correspondiente reacción del sistema organizativo, cuyo modularidad cumple los criterios de unidad del evento comunicativo.

  Cada evento comunicativo se representa como un rectángulo redondeado y se le da un identificador y un nombre descriptivo. El identificador sirve para propósitos de trazabilidad y suele ser un código compuesto de un mnemotécnico (relacionado con el sistema de organización o el proceso comercial al que se atribuye el evento) y un número (por ejemplo, VENTA 1, donde VENTA es el acrónimo del proceso de gestión de ventas y 1 es sólo un número que debe ser único dentro de este proceso particular). El actor de la interfaz se especifica textualmente en la parte inferior del evento comunicativo

- **Un evento comunicativo especializado (Specialised communicative event):** indica que son posibles diferentes alternativas en el camino a través del proceso comercial. La especialización puede ser inducida externamente, ya sea por variantes estructurales o de dominio en la estructura del mensaje que se asocia al evento, o por una decisión explícitamente tomada por el actor principal.

- **Evento Variante (Event variants)**: Es referirse como variante del evento a cada comportamiento alternativo o cada una de las transiciones alternativas (si las hay) dentro de un evento comunicativo especializado.

- **Relación de precedencia (Precedence Relation)**: Una relación de precedencia entre dos eventos comunicativos (digamos, A y B) es una regla que consiste en una relación que define el tiempo relativo entre ellos. Indica que, para que se produzca el acontecimiento comunicativo B, A tiene que haber ocurrido necesariamente antes. Se dice que A es un "precedente directo" (evento comunicativo) de B; a la inversa, se dice que B es un "sucesor directo" (evento comunicativo) de A. La relación

de precedencia forma parte de la condición previa del evento comunicativo "sucesor directo".

- **Interacciones comunicativas entrante (Ingoing communicative interaction)**: alimentan principalmente la memoria del sistema de información con nueva información significativa. Por ejemplo, la realización de un pedido por parte de un cliente, la notificación de una avería en una máquina envasadora por parte de un operador, la decisión del director del Departamento de Gestión de Riesgos de aceptar una inversión. Estas interacciones suelen estar respaldadas (aunque no siempre) por formularios comerciales.

- **Interacciones comunicativas salientes (Outgoing communicative interaction)**: consultan principalmente la memoria del sistema de información. Estas interacciones se materializan a menudo en indicadores, listados e impresiones comerciales. Por ejemplo, la lista de clientes con una deuda superior a 6000 euros, un cuadro de flujo de caja, un recibo, una lista de nóminas.

El diagrama de eventos comunicativos representa las relaciones de precedencia entre los eventos comunicativos, los actores primarios de las interacciones comunicativas entrantes y (opcionalmente) los actores receptores de las interacciones comunicativas salientes y los actores de apoyo involucrados.

b) Plantilla de especificación de eventos: Una técnica de modelado destinada a especificar los requisitos asociados a un determinado evento comunicativo.

Los eventos comunicativos que aparecen en el Diagrama de Eventos Comunicativos deben ser descritos en detalle. Los requisitos asociados a un evento pueden estructurarse mediante una plantilla de especificación de eventos. La plantilla se compone de un encabezamiento, requisitos de contacto, requisitos de mensaje y requisitos de reacción.

El encabezamiento contiene información general sobre el evento comunicativo, es decir, el identificador del evento, su nombre, una descripción narrativa y, opcionalmente, un diagrama explicativo.

- **Identificador de eventos**: La identificación de eventos debe mantenerse consistente a lo largo de todo el análisis y la especificación del diseño para mejorar la trazabilidad de los requisitos. De esta manera, el identificador de eventos no necesita coincidir con el del diagrama de eventos

comunicativos. Normalmente consta de la sigla del proceso empresarial al que pertenece el evento y un número secuencial.

- **Nombre del evento:** El nombre del evento comunicativo debe indicar claramente el cambio en el sistema de sujeto que se está reportando en el evento.

- **Descripción narrativa:** Dado que las especificaciones de los requisitos tienen por objeto, en primer lugar, facilitar la comprensión del problema, se aconseja encarecidamente una descripción narrativa del acontecimiento.

- **Diagrama explicativo:** Además, cuando el evento sea complejo, se incluirá un diagrama explicativo que ilustre el flujo de tareas asociado. En dicho diagrama se recomienda centrarse únicamente en los acontecimientos comunicativos físicos, es decir, las acciones relacionadas con la adquisición, recodificación y distribución de la información que cumplan con la unidad de activación y los criterios de reacción.

Los requisitos de contacto están relacionados con las condiciones necesarias para establecer la comunicación. Por ejemplo, el actor principal, los posibles canales de comunicación (por ejemplo, fax, correo electrónico, en persona), la disponibilidad y las limitaciones temporales (por ejemplo, el horario de oficina para la recepción de pedidos), los requisitos de autenticación (por ejemplo, en España, los trámites burocráticos suelen exigir la presentación de un documento de identidad).

- **Actor principal:** El rol organizativo que es responsable de comunicarse con el sistema de información para comunicar un cambio en el sistema en cuestión.

- **Actores de apoyo:** Los roles organizativos que participan en las transferencias de mensajes, pero no aportan nueva información; es decir, los actores responsables de los eventos comunicativos físicos.

- **Actores de interfaz:** Los roles organizativos que se encargan de editar los mensajes de entrada en el formulario y códigos requeridos por el sistema de información. En un sistema de información en papel, el actor de interfaz es el que rellena los formularios comerciales en papel; en un sistema de información informatizado, el actor de interfaz es el que interactúa con la interfaz de usuario de la aplicación informática.

- **Requisitos de disponibilidad:** Se refieren al grado en que el sistema de información está en condiciones de participar en la interacción comunicativa en curso.

- **Requisitos medios:** Se refieren a la tecnología (esto incluye los formularios en papel) que soporta la interacción comunicativa en curso. En caso de que se disponga de formularios empresariales escaneados o de capturas de pantalla de una aplicación de software anterior, pueden incluirse en esta sección los que se aplican al evento comunicativo (por ejemplo, los que apoyan la interacción comunicativa en curso). En caso de que se trata de catálogos en un documento o repositorio diferente, se puede incluir una referencia a ellos.

- **Requisitos de acreditación:** Se refieren a los protocolos que el sistema organizativo prescribe para cada actor que participa en la interacción comunicativa en curso (es decir, informalmente, cómo saber que los actores son quienes dicen ser).

- **Requisitos de verificación:** Se refieren a:

  a) Garantizar que la documentación aportada (si la hay) no es fraudulenta.

  b) Confirmar la entrada de elementos físicos asociados a la interacción comunicativa en curso (por ejemplo, la entrada de existencias en un almacén).

Los requisitos de mensajes especifican el mensaje transmitido por el actor principal al sistema de información en un evento comunicativo y las restricciones relacionadas (por ejemplo, fiabilidad: certificar que un diploma proporcionado por un estudiante no es fraudulento). En cuanto al mensaje, tanto los aspectos metalingüísticos (por ejemplo, la estructura de los campos del mensaje estructura de los campos del mensaje, obligatoriedad de los campos) como los aspectos lingüísticos (por ejemplo, dominios de los campos valores de ejemplo) deben especificarse.

- **Aspectos metalingüísticos:** Se refieren a la estructura del mensaje, su edición y su visualización.

  - **Estructura del mensaje** que se transmite al sistema de información; es decir, su composición en términos de subestructuras complejas y campos de mensaje.

- **Operación de adquisición:** Indica si el dato que contiene un campo es una información nueva para el sistema de la organización o simplemente una recuperación de la información previamente comunicada (esto incluye información derivada como los importes totales) es importante para analizar si el evento es realmente un evento comunicativo o debe ser descartado por no aportar información nueva y significativa.

- **Descripción de los campos:** Se debe proporcionar una descripción para que sea comprensible.

- **Aspectos lingüísticos:** Se refieren al contenido del mensaje y a su significado.

  - **Dominios de los campos de mensajes.** Durante el análisis, una orientación del campo contenido del campo (preferiblemente no un tipo de datos en lenguaje de programación de programación).

  - **Valor de ejemplo.** Uno o varios valores realistas que el campo puede contener aclarar su significado; deben ser proporcionados por los actores de la organización que participan en el análisis (es decir, usuarios representativos).

  - **Derivación de los valores del campo.** Algunos campos (e incluso algunas subestructuras complejas Algunas subestructuras complejas se derivan de información ya conocida. Especificando derivación puede hacerse textualmente o mediante fórmulas.

- **Restricciones del mensaje:** Como las restricciones sobre la estructura del mensaje, sobre los dominios de los campos del mensaje, etc.

Un evento comunicativo no puede entenderse plenamente hasta que se defina con detalle la estructura de su mensaje de entrada. La especificación con precisión de la estructura de un mensaje de evento obliga y ayuda a los analistas y usuarios a marcar adecuadamente el límite de un evento y su significado para la organización.

Los requisitos de reacción describen cómo reacciona el sistema de información al acontecimiento comunicativo (es decir, al mensaje transmitido). Normalmente, el sistema de información procesa y almacena la nueva información (actualizando la memoria del sistema), extrae todas las conclusiones necesarias que pueden

inferirse de los nuevos conocimientos, y pone los nuevos conocimientos y conclusiones a disposición de los actores correspondientes (distribuyendo la información a otros actores para que puedan actuar en consecuencia). Por tanto, esta categoría de requisitos incluye el tratamiento o procesamiento de la información y las interacciones comunicativas salientes generadas por el evento, entre otros requisitos.

- **Tratamientos:** Definen qué cambios se producen en el sistema de información como resultado del evento comunicativo (por ejemplo, qué procesamiento tiene lugar, qué información se almacena). Se trata de definir cómo se relaciona la adquisición de información con el modelo de datos. En algunos casos basta con indicar que la información se almacena. En otros casos, hay que realizar un procesamiento complejo y, por tanto, hay que describir un algoritmo con mayor o menor nivel de detalle (textualmente, con pseudocódigo, etc.).

- **Comunicaciones vinculadas:** Especifican a quién hay que comunicar la ocurrencia del evento comunicativo; es decir, qué roles organizativos (o actores organizativos específicos) necesitan conocer la ocurrencia del evento y su información asociada para poder tomar otras acciones (por ejemplo, tomar decisiones). En muchos casos, durante el análisis, puede bastar con indicar el actor al que hay que informar del suceso; en tales casos, las comunicaciones vinculadas pueden expresarse simplemente como interacciones comunicativas salientes en el diagrama de sucesos comunicativos. Si es necesario especificar más detalles sobre la comunicación, puede hacerse en esta sección (por ejemplo, forma de salida escaneada, sketch).

- **Comportamientos vinculados:** se refieren a cómo la ocurrencia de un evento comunicativo afecta a futuras ocurrencias de eventos. Por ejemplo, la información proporcionada en este evento comunicativo puede condicionar futuros comportamientos del sistema. Esto incluye reglas de negocio o condiciones complejas (por ejemplo, tablas de decisión) que determinan reacciones futuras dependiendo de los valores proporcionados en el evento comunicativo actual.

Teniendo en cuenta la siguiente Figura 3, se muestra en la Tabla 1, un ejemplo de plantilla de evento comunicativo.

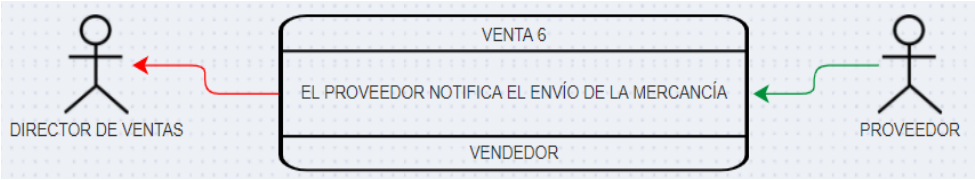

Figura 3: Diagrama de Evento Comunicativo Ejemplo Simple

VENTA 6. El proveedor notifica el envío de la mercancía

1. Información General

Objetivos

El objetivo de la organización es saber garantizar que la entrega se realiza a tiempo.

Descripción

Cuando el vehículo de transporte (normalmente un camión, pero a veces una furgoneta) recoge la mercancía del almacén del proveedor, éste llama por teléfono a la empresa para informar de que los envíos están de camino a sus destinos. El vendedor suele coger el teléfono y comunicar la información al jefe de ventas.

2. Requisitos de Contacto

Responsabilidades de los actores

- Actor principal: Proveedor
- Canal de comunicación: En persona
- Actor de apoyo: Vendedor

Requisitos temporales

- Frecuencia de aparición: 500 pedidos por semana

3. Requisitos de Comunicación

Estructura de Mensaje

CAMPO	OP	DOMINIO	VALOR DE EJEMPLO
DELIVERY NOTIFIC = < Order + Shipping timestamp >	 i i	 Client order date	 10352 05-09-2009 12:34

Campo	Descripción
Order	El pedido del cliente que ha sido enviado
Shipping timestamp	La fecha y hora en que el pedido ha sido enviado en el camión

4. Requisitos de Reacción
Tratamientos El pedido está marcado como "Entregado". **Comunicaciones vinculadas** El director de ventas debe conocer esta incidencia.

Tabla 1: Plantilla de Especificación de Evento Comunicativo - Ejemplo

c) Estructuras de mensaje: Una técnica de modelización para la especificación de los mensajes comunicados con (y dentro de) la organización.

Las Estructuras de Mensajes están compuestas por subestructuras, es decir, uno o más campos de datos, y agregaciones de subestructuras. Las agregaciones de las subestructuras son estructuras valiosas para la lógica empresarial, por lo que también se denominan Objetos de Negocio [2].

Es una técnica de especificación que permite describir, mediante un texto estructurado, el mensaje que se asocia a una interacción comunicativa. [5]

Subestructura				
Campo		**Subestructura compleja**		
Campo de datos	**Campo de Referencia**	**Subestructura de agregación**	**Subestructura de iteración**	**Subestructura de especialización**
a a representa un dato cuyo dominio es básico.	b b representa una referencia a un objeto de negocio ya conocido.	A=< a + b + C > A está compuesto por el campo de datos a, el campo de referencia b y la subestructura compleja C.	A={ a + b + C } A representa un conjunto de subestructuras que, a su vez, están compuestas por el campo de datos a, el campo de referencia b y la subestructura compleja C.	A=[a | b | C] A se compone del campo de datos a, o del campo de referencia b, o de la subestructura compleja C, exclusivamente.

Tabla 2: Resumen de las construcciones gramaticales de la técnica de modelado de estructuras de mensajes
[5]

CAMPO	OP	DOMINO	VALORES DE EJEMPLO
ORDER =			
< Order number +	g	number	10352
Request date +	i	date	31-08-2009
Payment type +	i	text	Cash
Client +	i	Client	56746163-R, John Papiro Jr.
DESTINATIONS =			
{ DESTINATION =			
< Address +	i	Client address	Blvd. Blue mountain, 35-14A, 2363 Toontown
Person in charge +	i	Text	Brayden Hitchcock
LINES =			
{ LINE =			
< Product +	i	Product	ST39455, Rounded scissors (cebra) box-100
Price +	i	money	25,40 €
Quantity >	i	number	35
}			
>			
}			
>			

Tabla 3: Ejemplo de Estructura de Mensaje
[5]

Se llama subestructura a un elemento constituido por campos o subestructuras complejas. Llamamos subestructura inicial a la subestructura que constituye el primer nivel de una estructura de mensaje. Por ejemplo, ORDER=<Order number+Request date+Payment type+Client+DESTINATIONS>.

Teniendo en cuenta la Tabla 3, la sintaxis de las estructuras de mensajes puede ser descrita en términos de los siguientes constructores gramaticales.

Campo (Field): Se trata de un elemento informacional básico del mensaje; aquel que no está compuesto de otros elementos. Existen dos tipos de campos:

- **Campo de datos (Data field)**: Especifica un campo que representa un dato cuyo dominio es básico. Por ejemplo, Payment type es un campo de datos textual, de la estructura de mensaje ORDER.

- **Campo de referencia (Reference field):** Especifica un campo cuyo dominio es un objeto de negocio. Por ejemplo, Client referencia a un cliente ya conocido por el SI.

Subestructura compleja (Complex substructure): Se trata de cualquier subestructura que tiene una composición interna. Existen tres tipos de subestructura compleja.

- **Subestructura de agregación (Aggregation substructure)**: Especifica una composición de varias subestructuras de manera que quedan agrupadas. Se representa mediante paréntesis angulares <>. Por ejemplo, LINE=<Product+Price+Quantity> especifica que una línea de pedido consiste de información sobre un producto, su precio, y la cantidad solicitada por el cliente.

- **Subestructura de iteración (Iteration substructure)**: Especifica un conjunto o repetición de aquellas subestructuras que contiene. Se representa mediante llaves { }. Por ejemplo, un pedido puede tener varios destinos y, para cada destino, se define un conjunto de líneas de pedido. Tanto DESTINATIONS como LINES son subestructuras de iteración.

  LINES={LINE=<Product+Price+Quantity>}

- **Subestructura de especialización (Specialisation substructure):** Especifica una o más variantes; es decir, alternativas estructurales. No hay ningún ejemplo de subestructura de especialización en la Tabla 3.

Para caracterizar un campo, se especifican las siguientes propiedades.

- **Nombre:** Cada campo debe tener un nombre significativo (v.g. Request date).

- **Operación**: Especifica la procedencia de la información que representa el campo.
  - **Introducción i**. La información del campo la provee el actor primario.

  - **Generación g**. La información del campo puede ser automáticamente generada por el SI.

  - **Derivación d**. La información del campo puede ser derivada de la memoria del SI porque ya se conoce; es decir, fue comunicada en un suceso comunicativo anterior. La derivación lleva asociada una fórmula de derivación.

- **Dominio**: Especifica el tipo de información que contiene el campo.

- **Ejemplo**: Un ejemplo de valor para el campo, aportado por la organización.

Hoy en día, el método cuenta con una herramienta para soportar el método, basada en Eclipse Modelling Framework (EMF).

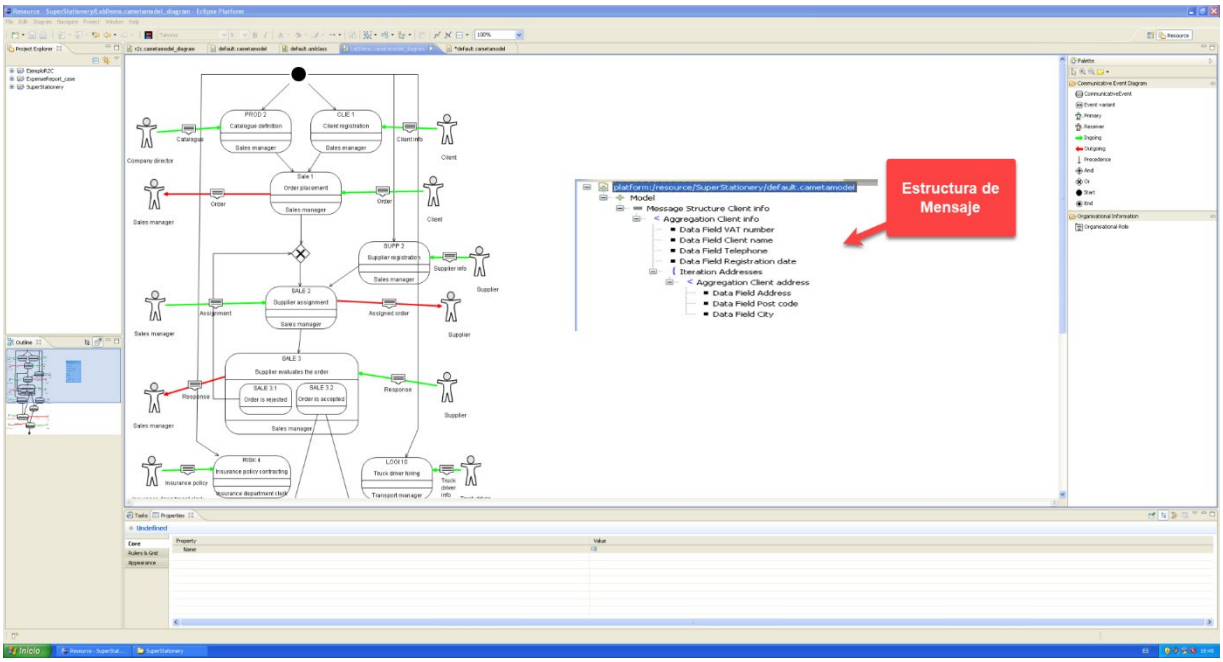

Figura 4: Herramienta GREAT (Eclipse Modelling Framework)

3.2. Método holístico de requisitos a código

El paradigma del Desarrollo Basado en Modelos tiene como objetivo representar a través de modelos todas las características del sistema. Esto lleva a trabajar con diferentes niveles de abstracción, donde cada nivel puede ser abordado con un método de modelación diferente Figura 5, que tiene varios modelos para representar las vistas del sistema. Sin embargo, no se aborda la calidad de la trazabilidad y las transformaciones entre esos niveles, desde el más abstracto hasta las generaciones de códigos [2].

En [2] se propone un método holístico para generar código a partir de diferentes niveles de abstracción (desde el espacio del problema hasta el espacio de la solución). Se centra en las transformaciones de la capa estratégica con el modelo i* a la capa de procesos empresariales con el modelo de Análisis Comunicacional (AC) y las transformaciones de la capa de procesos empresariales a la capa de modelos conceptuales de sistemas con el método OO, que puede generar automáticamente sistemas totalmente funcionales. Este paper es el primero que trata de la perspectiva de reunir todos los modelos en un único método de desarrollo.

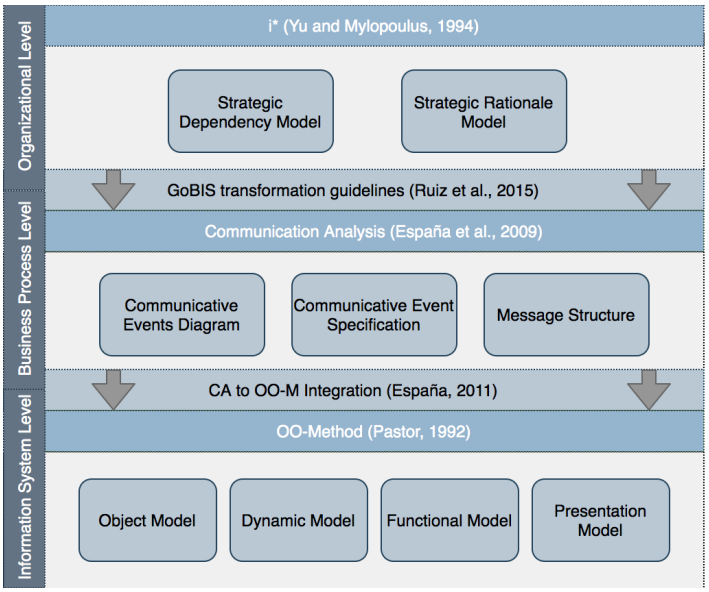

Figura 5: Propuesta de relación de los métodos de modelización [2]

En la Figura 5, se puede visualizar los tres métodos de modelización y la herramienta que lo soporta, modelado organizacional con i*, modelado de procesos de negocios con Análisis de Comunicación (CA), y modelado de sistemas con el Método OO que apoya la generación de código del sistema de información; sin embargo, estas herramientas son independientes por lo que se plantea demostrar que es posible diseñar un método de producción de software holístico que conecte con un sólido fundamento metodológico, los objetivos y requisitos de los interesados con su código asociado.

3.3. Definición del problema

Hoy en día, el modelado de procesos de negocios a través de Análisis Comunicacional cuenta con una herramienta para soportar el método, basada en Eclipse Modelling Framework (EMF). Sin embargo, tanto las limitaciones propias de EMF como la necesidad de integración de los modelos de AC con otros modelos, plantea el desafío de diseñar e implementar una nueva herramienta en entorno web para facilitar su interoperabilidad con otras herramientas de modelado.

3.4. Objetivos

3.4.1. Objetivo General

Diseñar e implementar una herramienta de soporte para el modelado de procesos de negocio con Análisis Comunicacional.

3.4.2. Objetivos Específicos

- Analizar los requerimientos funcionales y no funcionales de la herramienta.
- Diseñar la herramienta para la satisfacción de los requerimientos funcionales y no funcionales de la herramienta.
- Implementar la herramienta de modelado de Análisis Comunicacional.
- Verificar y validar la implementación de la herramienta a través de un cuestionario.

Capítulo 4

Requisitos de la Solución

4.1. Solución actual (GREAT)

El grupo de investigación en Ingeniería de Requisitos del Centro de Investigación PROS (UPV) ha creado GREAT Process Modeller, un prototipo de herramienta para modelar los procesos empresariales y especificar los requisitos de los sistemas de información, con el fin de captar las necesidades de la organización de forma ágil, reduciendo al mismo tiempo la duración del desarrollo. (GREAT Process Modeller: Global Reengineering Environment with Automated Transformations)

GREAT Process Modeller está concebido para ser utilizado de la siguiente manera: El analista modela con agilidad los procesos comerciales empleando el editor gráfico y las plantillas de requisitos asociadas, en colaboración con los interesados.

GREAT Process Modeller es la herramienta que soporta el método y se basa en las tecnologías más avanzadas para la construcción de herramientas de desarrollo basadas en modelos (Eclipse EMF, GMF, Xtext).

Sin embargo, cuenta con limitaciones como la necesidad de integración de los modelos de Análisis Comunicacional con otros modelos.

4.2. Una arquitectura distribuida para la transformación de modelos

En el área de la ingeniería del software, el desarrollo dirigido por modelos (MDD) es un paradigma de desarrollo de software donde los principales artefactos de software son modelos, a partir de los cuales se puede generar código y otros artefactos. Dejando la implementación solo como un requerimiento técnico. Dado esto se han desarrollado variadas técnicas de modelados que atacan el problema a distintos niveles de abstracción, pero sus herramientas no se comunican entre ellas, y no son capaces de cumplir el principal objetivo del desarrollo dirigido por modelos que es la autogeneración del código.

Existen herramientas aisladas entre ellas, que no cumplen los objetivos del desarrollo guiado por modelos. Por ende, no existe la comunicación necesaria para la generación de una herramienta que de soporte al desarrollo dirigido por modelos a un alto nivel de abstracción.

En el trabajo de Título Diseño y desarrollo de un sistema de Transformación de modelos basado en Web, se busca generar una arquitectura que permita la comunicación de estas técnicas y sus transformaciones necesarias para dejar un flujo que permita desde un alto nivel autogenerar código. [7]

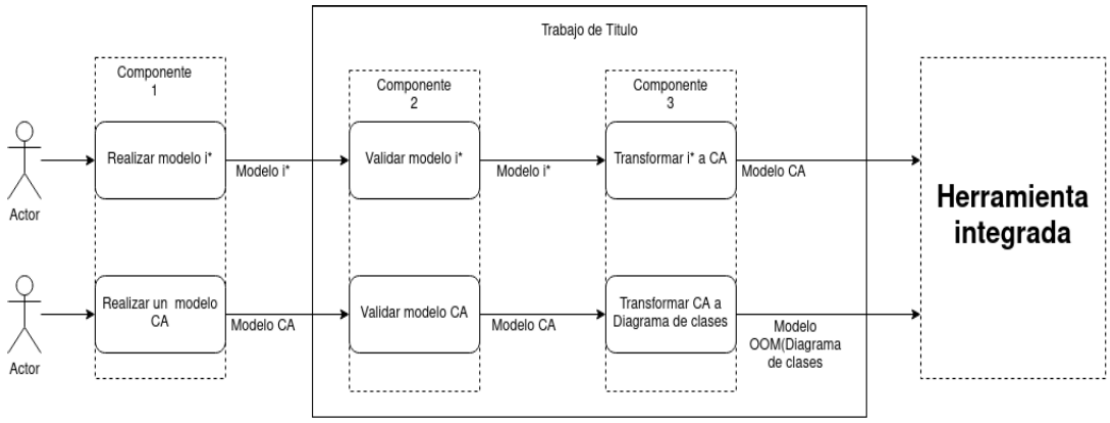

Figura 6: Diagrama de la arquitectura
[7]

Cabe mencionar que el componente 1 y la herramienta integrada se dan por creados.

4.3. Requerimientos

4.3.1. Requerimientos funcionales

Los requisitos funcionales son los que especifican el manejo que requiere el sistema y las funciones que debe incluir.

- RF01: La herramienta permitirá realizar diagramas de eventos comunicativo.
- RF02: La herramienta permitirá agregar estructuras de mensajes en el diagrama de eventos comunicativos.
- RF03: La herramienta permitirá guardar el diagrama de evento comunicativo en formato JSON.
- RF04: La herramienta permitirá abrir un diagrama previamente guardado en formato JSON.
- RF05: La herramienta permitirá validar las relaciones entre elementos indicando si a algún elemento le falta relaciones.

- RF06: Las interacciones comunicativas de entrada y salida, como la relación de precedencia se encuentran identificadas con color.

4.3.2. Requerimientos no funcionales

- RNF01: La herramienta se desarrollará con mxGraph y React bajo el IDE Visual Studio Code.

- RNF02: Al ser una herramienta que se visualiza a través de un navegador será multiplataforma.

- RNF03: Al ser una herramienta en la web, será necesaria conectividad a la red.

- RNF04: La herramienta debe ser fácil de usar.

Capítulo 5

Propuesta de solución

En este capítulo se aborda el diseño e implementación de la herramienta de soporte para el modelado de procesos de negocio con Análisis Comunicacional.

5.1. Diseño de interfaz de usuario

La interfaz de usuario principal está compuesta por tres (03) componentes (sidebar, toolbar y editor) cuya distribución se muestra en la Figura 7. La interfaz de usuario es intuitiva y fácil de entender para el usuario; está orientado a la usabilidad y visualización agradable.

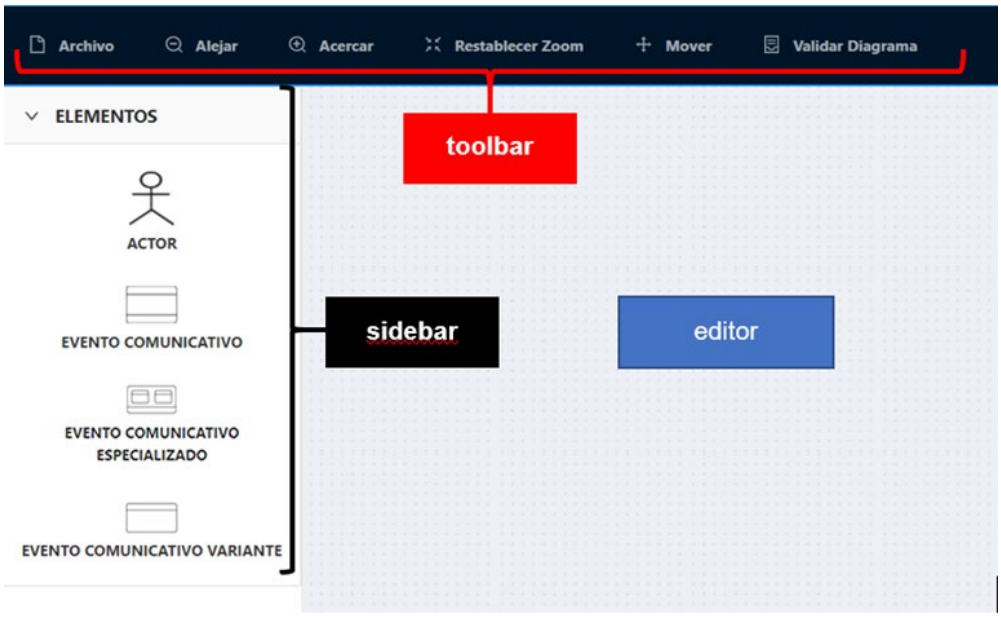

Figura 7: Pantalla Principal – Componentes

- **Toolbar**: El componente toolbar es la barra de herramientas que contiene los botones desde los cuales el usuario puede ejecutar diferentes funciones de la herramienta sobre el editor.

- **Sidebar**: El componente sidebar es la barra lateral que contiene los elementos desde los cuales el usuario puede arrastrar hacia el editor.

- **Editor**: El componente editor es el principal componente de la herramienta que integra a los dos (02) componentes mencionados (toolbar y sidebar). Este componente sirve como una especie de pizarra donde los elementos de sidebar que al ser arrastrado y soltado sobre el componente se va dibujar según la forma seleccionada; a su vez, estos pueden moverse o eliminarse.

En la Figura 8, se puede observar los diferentes elementos que han sido arrastrados sobre el editor y las relaciones a través de flechas. Las flechas de color verde indica la interacción comunicativa entrante, la roja interacción comunicativa saliente y la negra relación de precedencia.

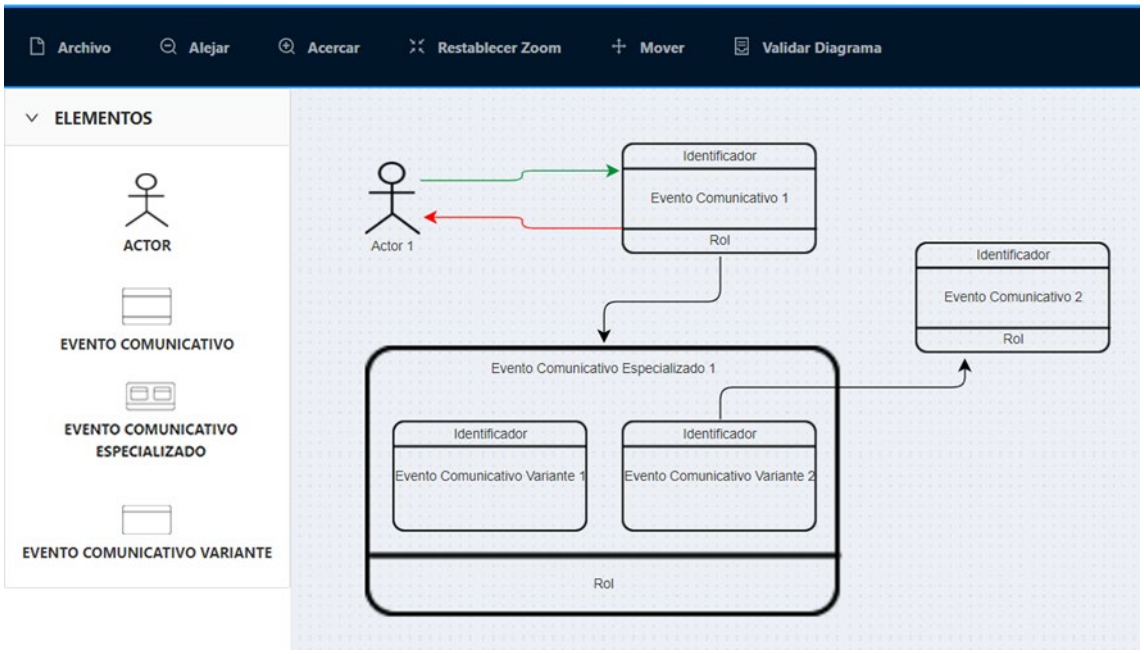

Figura 8: Diagrama de Evento Comunicativo – Herramienta Creada

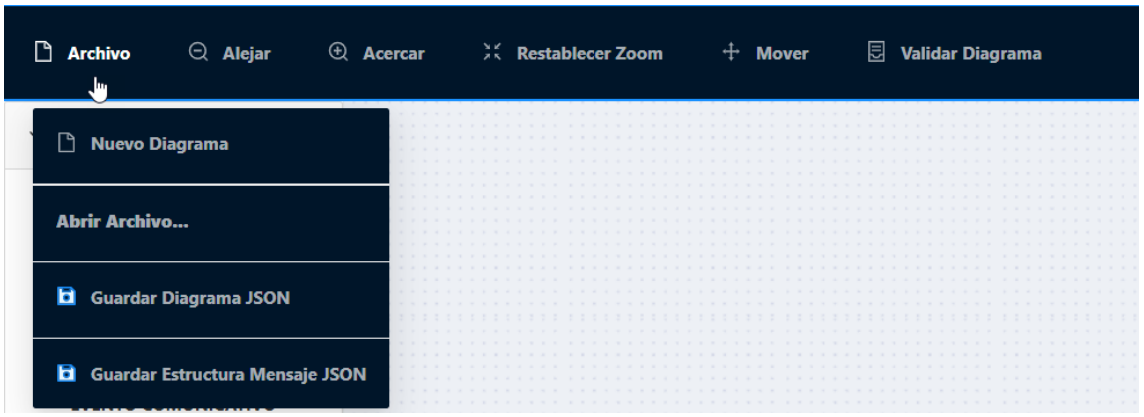

Figura 9: Elementos del componente toolbar

En la Figura 9, se presenta los elementos del componente toolbar que está compuesto por un menú que se detalla a continuación:

Menú Archivo: Este menú está compuesto por sub menús:

Nuevo Diagrama: Permite limpiar el editor e iniciar con el nuevo diagrama a realizar.

Abrir Archivo: Permite abrir el archivo que tengamos guardado con la misma herramienta y poder volver a visualizarlo en el editor y poder seguir realizando cualquier acción sobre ello.

Guardar Diagrama JSON: Permite guardar el diagrama realizado en formato JSON.

Guardar Estructura Mensaje JSON: Permite guardar solo la estructura de mensaje en formato JSON.

Menú Alejar: Permite disminuir el zoom del editor.

Menú Acercar: Permite aumentar el zoom del editor.

Menú Restablecer Zoom: Permite regresar al zoom por defecto que se visualizar al ingresar a la herramienta.

Menú Mover: Permite poder deslizarse sobre el editor.

Menú Validar Diagrama: Permite validar el diagrama con respecto a las relaciones de los elementos; como puede visualizarse en la Figura 10, que se encuentra ambos elementos relacionados por lo tanto es correcto.

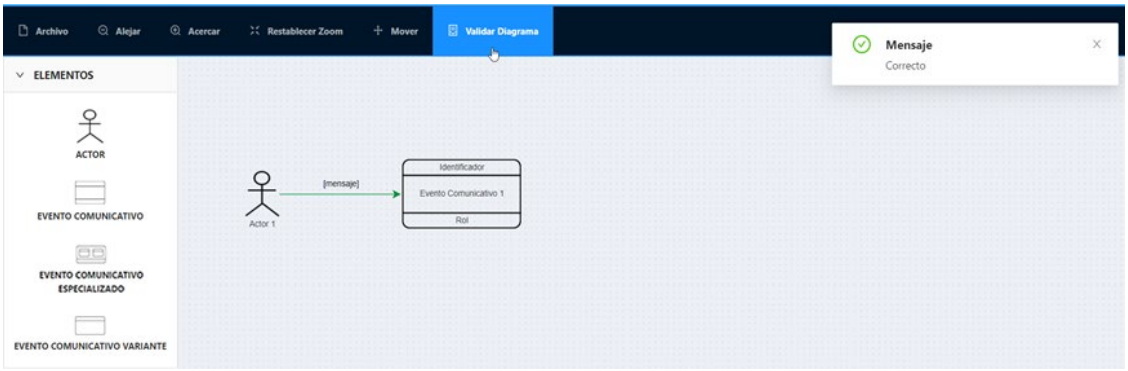

Figura 10: Menú Validar Diagrama - Correcto

En la Figura 11, al no encontrarse ambos elementos relacionados se muestra las advertencias necesarias al usuario para la verificación correspondiente.

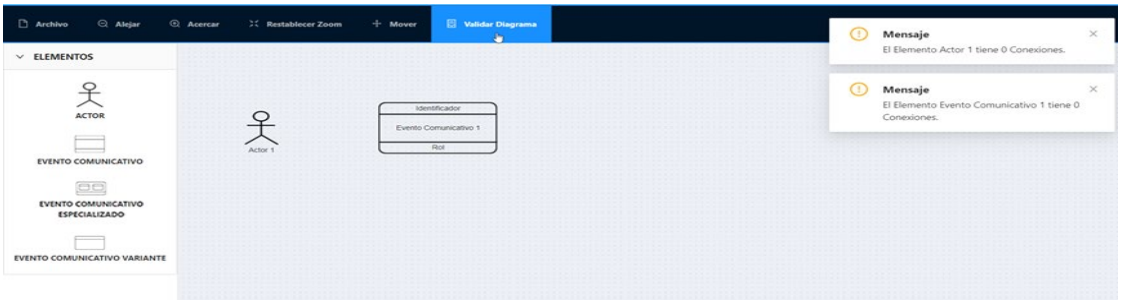

Figura 11: Menú Validar Diagrama - Advertencias

En la Figura 12, se puede visualizar la primera forma de poder editar los valores de un elemento que se ejecuta al hacer doble clic sobre el elemento para que entre a modo edición.

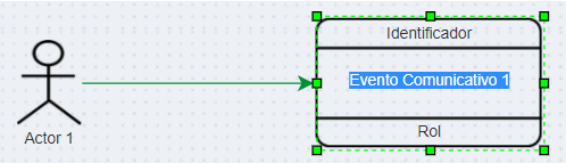

Figura 12: Editar valores elemento - Forma 1

En la Figura 13, se puede visualizar la segunda forma de poder editar los valores de un elemento que se ejecuta al hacer clic derecho sobre el elemento para que se muestre una ventana modal para el ingreso de los nuevos datos como se visualiza en la Figura 14.

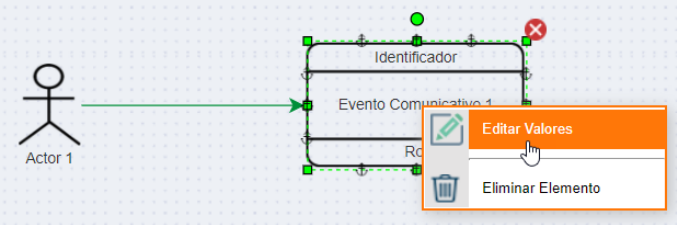

Figura 13: Editar valores elemento - Forma 2

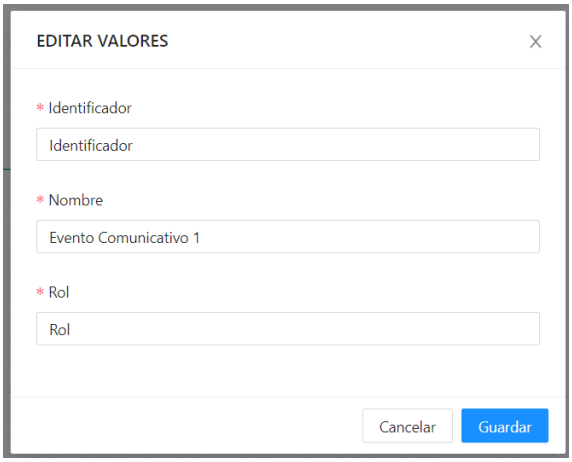

Figura 14: Ventana modal edición de valores

En la Figura 15, se puede visualizar las dos (02) formas de poder eliminar un elemento ya se a través del menú mostrado o a través del icono X. Para ambas formar antes de eliminar un elemento se muestra una alerta de confirmación del elemento

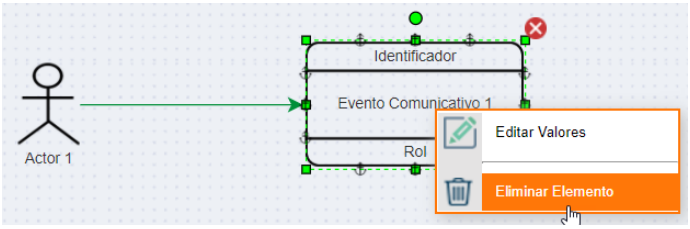

Figura 15: Eliminar Elemento

En la Figura 16, al hacer clic derecho sobre la flecha se muestra la opción para agregar la estructura de mensaje correspondiente.

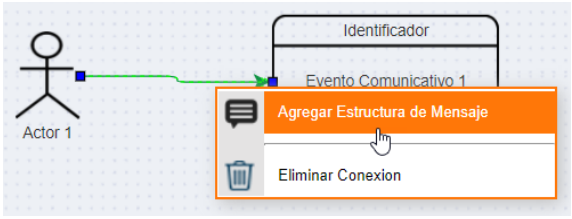

Figura 16: Agregar Estructura de mensaje

Al realizar dicha acción se muestra una como se muestra en la Figura 17, con los respectivos campos y subestructura compleja.

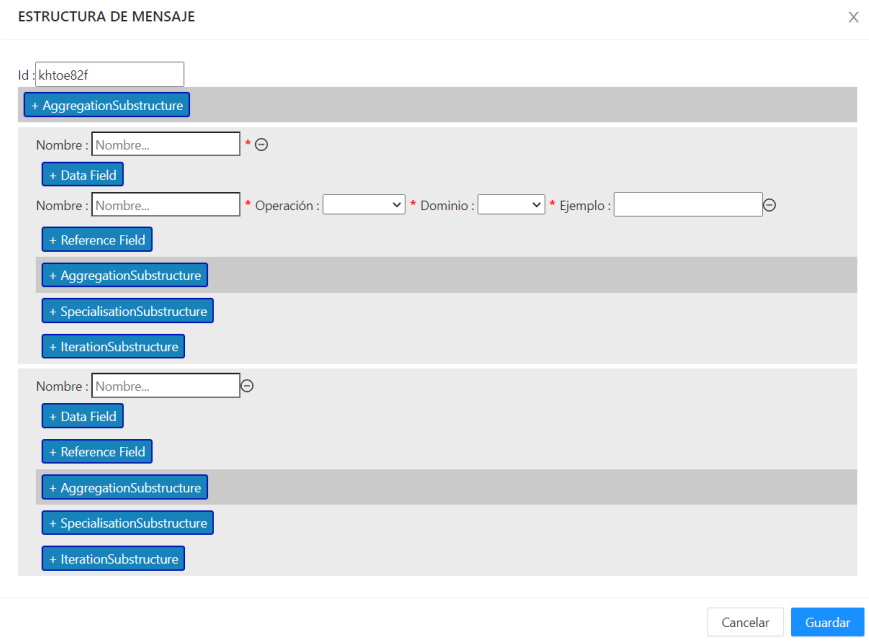

Figura 17: Ventana modal estructura de mensaje

En la Figura 18, con la finalidad de poder orientar al usuario una vez agregado la estructura de mensaje se coloca sobre la flecha un texto ([mensaje]) indicado que dicha relación ya se encuentra con una estructura de mensaje.

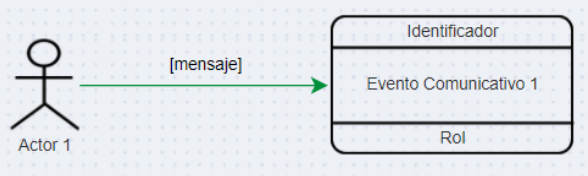

Figura 18: Relación con estructura de mensaje

Por último, se añadió una funcionalidad a la herramienta para evitar que cuando se quiera cerra la pestaña del navegador donde se está trabajando se alerte al usuario primero y no perder el trabajo realizado como se visualiza en la Figura 19.

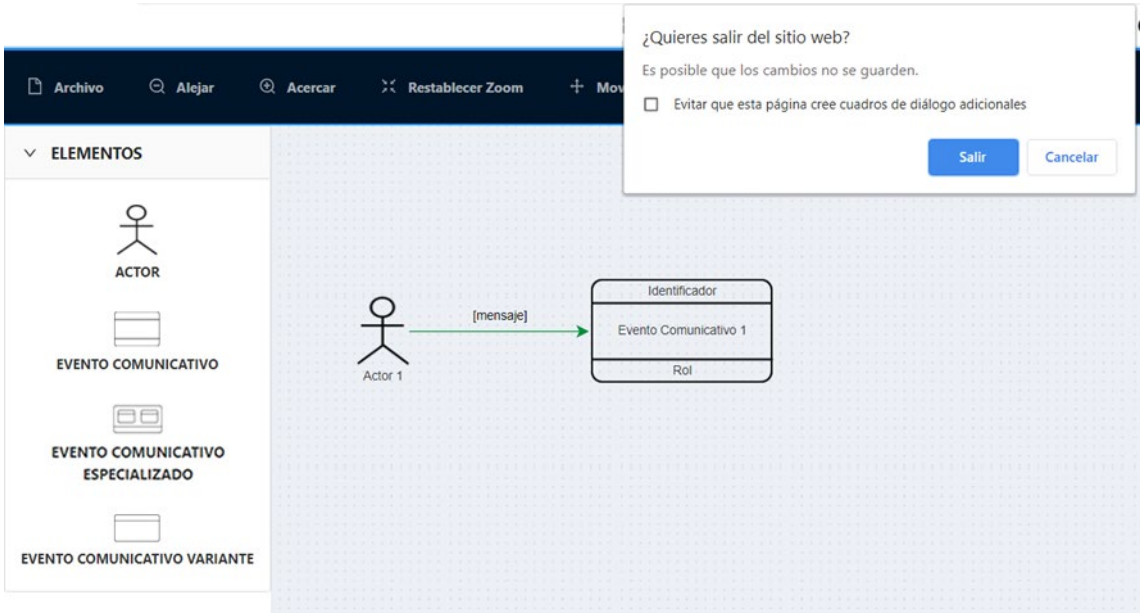

Figura 19: Alerta antes de cerrar la herramienta

5.2. Diseño de persistencia (JSON)

JSON (JavaScript Object Notation - Notación de Objetos de JavaScript), se trata de un formato para guardar e intercambiar información que cualquier persona pueda leer. Los archivos JSON contienen solo texto y usan la extensión ".json".

JSON es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo [8]. JSON está constituido por dos estructuras:

- Una colección de pares de nombre/valor. En varios lenguajes esto es conocido como un objeto, registro, estructura, diccionario, tabla hash, lista de claves o un arreglo asociativo.

- Una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementa como arreglos, vectores, listas o secuencias.

Para ejemplificar, supongamos que queremos enviar datos de unos clientes, y necesitamos transferir esta información a un servidor o una aplicación web. Esto puede lograrse fácilmente con JSON como se puede visualizar en su formato abajo [9].

```
{"clientes": [
    {"primerNombre": "Marcos", "apellido": "Perez"},
    {"primerNombre": "Andres", "apellido": "Lopez"},
    {"primerNombre": "Jose", "apellido": "Flores"}
]}
```

En el caso de la herramienta que se ha implementado cada elemento guarda valores y en el caso de la relación de dos elementos se puede añadir una estructura de mensaje; teniendo en cuenta que la persistencia es la acción de preservar la información de un objeto de forma permanente (guardado), pero a su vez también a poder recuperar la información del mismo (leerlo) para que pueda ser nuevamente utilizado.

Es por el cual la herramienta cuenta con la funcionalidad de poder guardar y leer la información de los diagramas realizados; dicha información se almacena en un formato ligero y entendible (JSON) para el usuario.

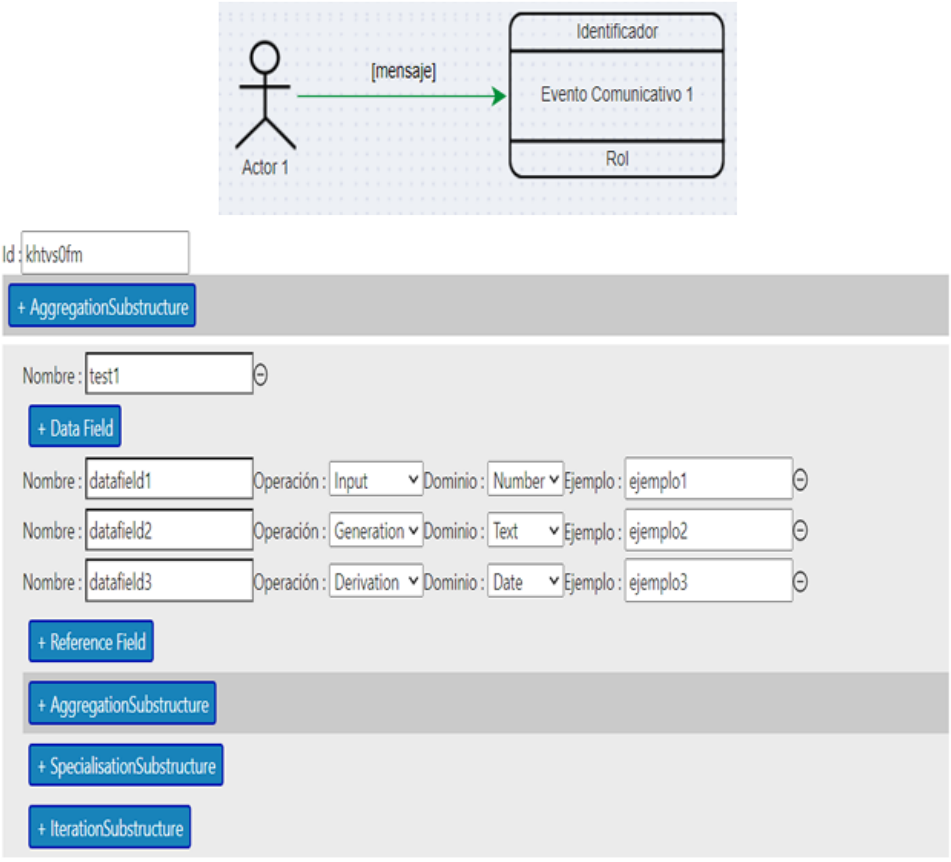

Figura 20: Ejemplo de diagrama con estructura de mensaje

En la Figura 20, se puede visualizar un ejemplo de diagrama con su respectiva estructura de mensaje que, al ser guardada por el usuario, genera en un archivo con extensión ".json" con la siguiente información que representa todo lo que se encuentra en el editor.


```
{
   "mxGraphModel": {
      "root": {
         "mxCell": [
            {
               "_attributes": {
                  "id": "khtvo1by"
               }
            },
            {
               "_attributes": {
                  "id": "khtvo1bz",
                  "parent": "khtvo1by"
               }
            },
            {
               "_attributes": {
                  "id": "khtvrmpx",
                  "value": "Actor 1",
                  "style":
"html=1;verticalLabelPosition=bottom;fontColor:#fff;verticalAlign=top;imageAspect=0;image=analisis_comunicacional/images/actor.png",
                  "vertex": "1",
                  "parent": "khtvo1bz",
                  "shapeName": "actorac"
               },
               "mxGeometry": {
                  "_attributes": {
                     "x": "67.5",
                     "y": "162",
                     "width": "45",
                     "height": "60",
                     "as": "geometry"
                  }
               }
            },
            {
               "_attributes": {
                  "id": "khtvrpvx",
                  "value": "Evento Comunicativo 1",
                  "style":
"html=1;align=center;verticalAlign=middle;whiteSpace=wrap;image=analisis_comunicacional/images/communicative_event.png",
                  "vertex": "1",
                  "parent": "khtvo1bz",
                  "shapeName": "cevent"
               },
               "mxGeometry": {
                  "_attributes": {
                     "x": "270",
                     "y": "145",
                     "width": "160",
                     "height": "94",
                     "as": "geometry"
                  }
               }
```

```
            },
            {
               "_attributes": {
                  "id": "khtvrpw2",
                  "value": "Identificador",
                  "style": "deletable=0;movable=0",
                  "vertex": "1",
                  "parent": "khtvrpvx"
               },
               "mxGeometry": {
                  "_attributes": {
                     "y": "12",
                     "width": "160",
                     "as": "geometry"
                  }
               }
            },
            {
               "_attributes": {
                  "id": "khtvrpw5",
                  "value": "Rol",
                  "style": "deletable=0;movable=0",
                  "vertex": "1",
                  "parent": "khtvrpvx"
               },
               "mxGeometry": {
                  "_attributes": {
                     "y": "81",
                     "width": "160",
                     "as": "geometry"
                  }
               }
            },
            {
               "_attributes": {
                  "id": "khtvs0fm",
                  "value": "[mensaje]\n\n",
                  "style": "exitX=1;exitY=0.5;entryX=0;entryY=0.5;",
                  "edge": "1",
                  "parent": "khtvo1bz",
                  "source": "khtvrmpx",
                  "target": "khtvrpvx"
               },
               "mxGeometry": {
                  "_attributes": {
                     "relative": "1",
                     "as": "geometry"
                  }
               },
               "Object": {
                  "_attributes": {
                     "id": "khtvs0fm",
                     "as": "valueEstructuraMensaje"
                  },
                  "Object": {
                     "_attributes": {
                        "as": "agg_sub"
                     },
                     "Array": {
                        "_attributes": {
                           "as": "agg_sub"
                        },
                        "Object": {
```

```
                        "_attributes": {
                            "nombre": "test1"
                        },
                        "Array": {
                            "_attributes": {
                                "as": "data_field"
                            },
                            "Object": [
                                {
                                    "_attributes": {
                                        "nombre": "datafield1",
                                        "operacion": "Input",
                                        "dominio": "Number",
                                        "ejemplo": "ejemplo1"
                                    }
                                },
                                {
                                    "_attributes": {
                                        "nombre": "datafield2",
                                        "operacion": "Generation",
                                        "dominio": "Text",
                                        "ejemplo": "ejemplo2"
                                    }
                                },
                                {
                                    "_attributes": {
                                        "nombre": "datafield3",
                                        "operacion": "Derivation",
                                        "dominio": "Date",
                                        "ejemplo": "ejemplo3"
                                    }
                                }
                            ]
                        }
                    }
                }
            }
        }
    ]
  }
 }
}
```


5.3. Diagrama de componentes

El diagrama de componente proporciona una vista de alto nivel de los componentes dentro de la herramienta implementada.

Los diagramas de componentes muestran como el sistema está dividido en componentes y las dependencias entre ellos, además proveen una vista arquitectónica de alto nivel del sistema y ayuda a los desarrolladores a visualizar el mejor camino a la implementación [10].

La herramienta implementada se encuentra compuesto por los componentes que se reflejan en la Figura 21, la mayor parte se encuentra en el componente MyEditor que se encarga de otorgar la funcionalidad necesaria para la interacción con el usuario.

A continuación, se presenta, el diagrama componentes de la herramienta implementada y se describe algunas de ellas.

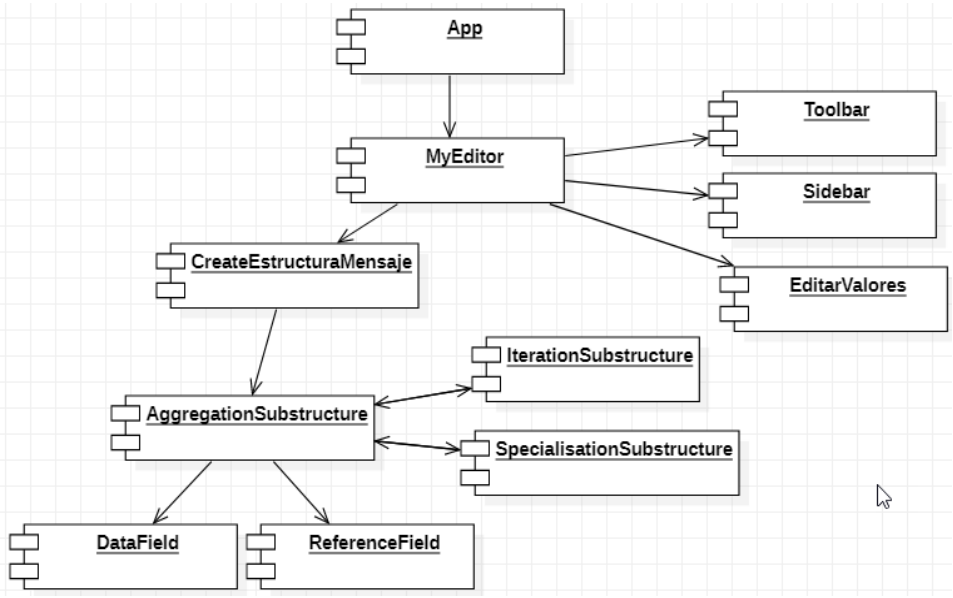

Figura 21: Diseño de Componentes

App: Componente principal (nodo raíz) encargado de renderizar en un archivo "index.html" todo el contenido que se encuentra en MyEditor.

MyEditor: Componente encargado de otorgar la funcionalidad necesaria al usuario.

Toolbar: Componente de la parte superior de la herramienta que está integrado por los menús que se muestra en la Figura 9.

SideBar: Componente de la parte lateral izquierda de la herramienta donde se encuentran los elementos que son parte del diagrama de eventos comunicacional.

CreateEstructuraMensaje: Componente que se encarga de mostrar a través de un formulario modal los campos correspondientes a la Estructura de mensaje compuesto por (Data field, Reference field, Aggregation substructure, Iteration substructure y Specialisation substructure), Figura 17.

5.4. Descripción de la solución

El diseño e implementación de la herramienta se ha realizado haciendo uso de las librerías JavaScript mxGraph y React con el editor de código fuente Visual Studio Code desarrollado por Microsoft; sin embargo, era necesario tener control del código fuente en un repositorio para por el cual se usó GitLab.

5.1.1. Librería JavaScript

JavaScript se presenta como un lenguaje de desarrollo de aplicaciones cliente/servidor a través de internet [11].

JavaScript es el lenguaje de programación más utilizado hoy. Junto a HTML y CSS, le da vida a la gran mayoría de sitios web que visitamos [12].

Las librerías JavaScript son un código reutilizable que a menudo proporcionan muchas funcionalidades para que los desarrolladores no tengan que preocuparse por muchas funciones; de esta manera, pueden usarlas para crear páginas web fácilmente.

5.1.2. mxGraph

Para poner en práctica la aplicación y sus funciones sin desarrollar métodos y características para dibujar y crear diagramas, es necesario utilizar una biblioteca que ya implementé estas funcionalidades. Para este propósito, se ha elegido mxGraph.

mxGraph, cuyo logotipo se muestra en la Figura 22, es un componente de JavaScript que proporciona características dirigidas a aplicaciones que muestran diagramas y gráficos interactivos.

Figura 22: Logotipo de mxGraph

Al ser una biblioteca para desarrolladores, mxGraph no está diseñada específicamente para proporcionar una aplicación lista para usar, si no que proporciona toda la funcionalidad comúnmente requerida para dibujar, interactuar y asociar un contexto con un diagrama.

MxGraph es una biblioteca JavaScript que permite integrar grafos y diagramas interactivos en las aplicaciones web. De esta forma, el usuario puede crear elementos gráficos e interactuar con ellos mediante la interfaz. Además, estos elementos pueden asociarse a un modelo de datos definido por el desarrollador, permitiendo realizar cambios en el modelo a través del diagrama [13].

Otra ventaja es que esta biblioteca es totalmente gratuita y libre. Gracias a su gran flexibilidad y robustez, mxGraph es utilizado por muchas

grandes empresas, por ejemplo, el servicio web Draw IO (Figura 23) está implementado con esta tecnología.

Figura 23: Logotipo de draw.io

Por otra parte, mxGraph proporciona una API completa que permite a los desarrolladores crear sus aplicaciones de una forma consistente y en cualquier entorno. También cabe mencionar que está biblioteca está implementada únicamente con JavaScript, sin bibliotecas de terceros, de forma que la compatibilidad con otros frameworks está asegurada. Esto resulta de vital importancia, pues en la elección de las tecnologías de desarrollo se ha tenido en cuenta la flexibilidad y compatibilidad entre éstas.

MxGraph utiliza un lenguaje de gráficos de vectores, normalmente SVG, y HTML para dibujar los diagramas, por esta razón puede ejecutarse en cualquier tipo de navegador.

MxGraph se estructura en 8 paquetes que engloban toda su funcionalidad y que se detallan a continuación.

- El paquete **«editor»** que proporciona las clases necesarias para crear un editor de diagramas.

- Los paquetes **«view»** y **«model»** que implementan todos los componentes necesarios para crear y dibujar los diferentes elementos de un diagrama.

- Los paquetes **«handler»**, **«layout»** y **«shape»** que integran los manejadores de eventos, la forma en la cual se distribuyen los elementos gráficos y las formas o figuras de éstos.

- El paquete **«util»** que implementa utilidades generales para los desarrolladores, como la capacidad de arrastrar y soltar unos elementos gráficos.

- El paquete **«io»** que proporciona diferentes clases para codificar y descodificar los diagramas en XML.

Como se ha podido observar, mxGraph proporciona todas las funcionalidades necesarias para crear diagramas de forma muy sencilla. El poder asociar datos a los elementos gráficos supone una gran ventaja para alcanzar los objetivos de este proyecto. Además, su gran flexibilidad permite integrarlo junto a otros frameworks, como React. Todas estas

razones, más el hecho de que sea totalmente gratuita, han propiciado la elección de esta tecnología para dibujar los diagramas. En la Figura 24, se puede visualizar un ejemplo simple de un diagrama.

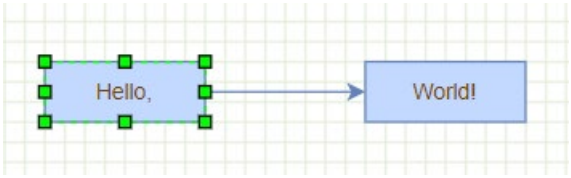

Figura 24: Ejemplo de un diagrama simple

5.1.3. React

React, cuyo logotipo se muestra en la Figura 25, es una biblioteca JavaScript para crear interfaces de usuario.

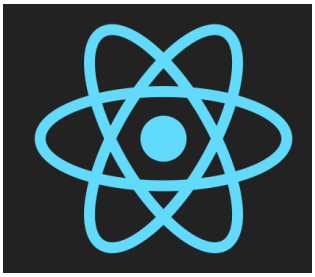

Figura 25: Logotipo de React

React es una biblioteca popular utilizada para crear interfaces. Fue construida en Facebook para abordar algunos de los desafíos asociados con los sitios web a gran escala, basados en datos. Cuando React fue lanzado en 2013, el proyecto fue visto inicialmente con cierto escepticismo porque las convenciones de React son bastante únicas [14].

Uno de los aspectos más importantes de React es el hecho de que puede crear componentes, que son como elementos HTML personalizados y reutilizables, para construir interfaces de usuario de manera rápida y eficiente.

React introdujo un estilo de programación declarativo, funcional y basado en componentes para crear interfaces de usuario interactivas para aplicaciones web principalmente de una sola página. React ofrece un renderizado ultrarrápido al hacer uso de 'Virtual DOM'[1] que renderiza solo aquellos componentes que han cambiado en lugar de renderizar toda la página. Otra característica clave de React es el uso de una sintaxis JSX[2] más simple en lugar de JavaScript.

[1] definición de Virtual DOM disponible en: **https://es.reactjs.org/docs/faq-internals.html**
[2] definición de JSX disponible en: **https://es.reactjs.org/docs/introducing-jsx.html**

React también simplifica la forma en que se almacenan y manipulan los datos, utilizando el **state** y los **props** (**props** se pasa al componente "similar a los parámetros de una función" mientras que **state** se administra dentro del componente "similar a las variables declaradas dentro de una función" [15].

```
class ParentComponent extends Component {
    render() {
        return (
            <ChildComponent name="First Child" />
        );
    }
}

const ChildComponent = (props) => {
    return <p>{props.name}</p>;
};
```

Figura 26: Ejemplo de Props

Props, según la Figura 26, en primer lugar, necesitamos definir / obtener algunos datos del componente principal y asignarlos al atributo "prop" de un componente secundario.

```
<ChildComponent name="First Child" />
```

"name" es un prop definido aquí y contiene datos de texto. Entonces podemos pasar datos con props como si estuviéramos dando un argumento a una función.

```
const ChildComponent = (props) => {
  // statements
};
```

Y finalmente, usamos la notación de puntos para acceder a los datos de la propiedad y representarlos:

```
return <p>{props.name}</p>;
```

State, según la Figura 27, a diferencia de los props, los componentes no pueden transmitir datos con estado, pero pueden crearlos y administrarlos internamente.

```
class Test extends React.Component {
    constructor() {
        this.state = {
            id: 1,
            name: "test"
        };
    }

    render() {
        return (
            <div>
              <p>{this.state.id}</p>
              <p>{this.state.name}</p>
            </div>
        );
    }
}
```

Figura 27: Ejemplo de State

El estado no debe modificarse directamente, pero puede modificarse con un método especial llamado **setState( )**.

```
this.state.id = “2020”; // wrong

this.setState({         // correct
    id: "2020"
});
```

Figura 28: Ejemplo Correcto e Incorrecto de State

Un cambio en el state ocurre según la entrada del usuario, desencadenando un evento, etc. Además, los componentes de React (con state) se renderizan en base a los datos en el state. El state contiene la información inicial.

Así que cuando el estado cambia, React es informado e inmediatamente vuelve a renderizar el DOM - no todo el DOM, sino sólo el componente con el estado actualizado. Esta es una de las razones por las que React es rápido.

React cuenta con el apoyo de una enorme comunidad de desarrolladores, abundantes recursos de aprendizaje y una adopción masiva de la industria en todos los rincones del mundo.

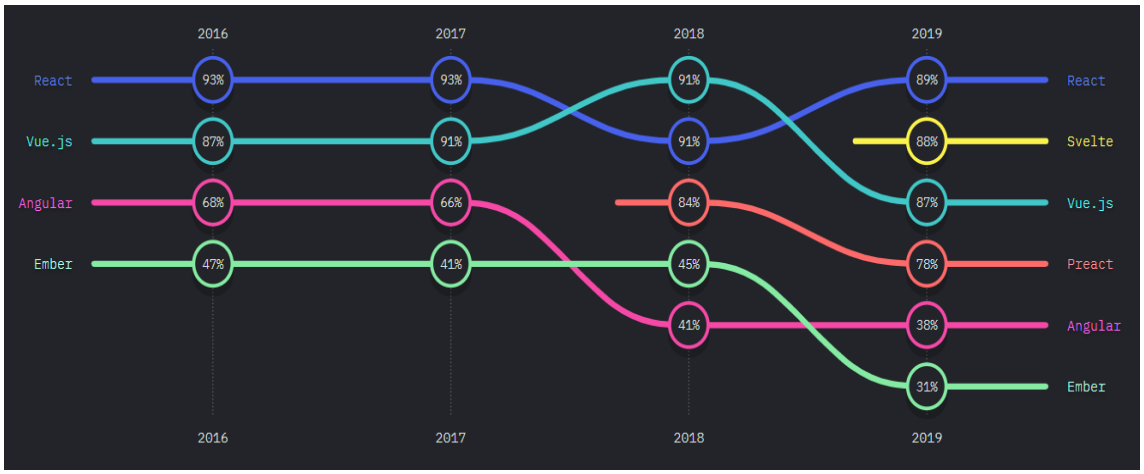

Figura 29: Front-end frameworks and libraries.
Tomado de [16]

React ha encabezado consistentemente las listas de popularidad en todo el marco de JavaScript, ya sea la encuesta de Stack overflow Developer Survey o la encuesta de State OF JS Survey. React ha ganado constantemente la corona como el marco favorito de JavaScript de primera línea. Las empresas y marcas más importantes del mundo, como Airbnb, Facebook, Instagram, Netflix, Twitter, WhatsApp y muchas otras, han sido creadas con React.

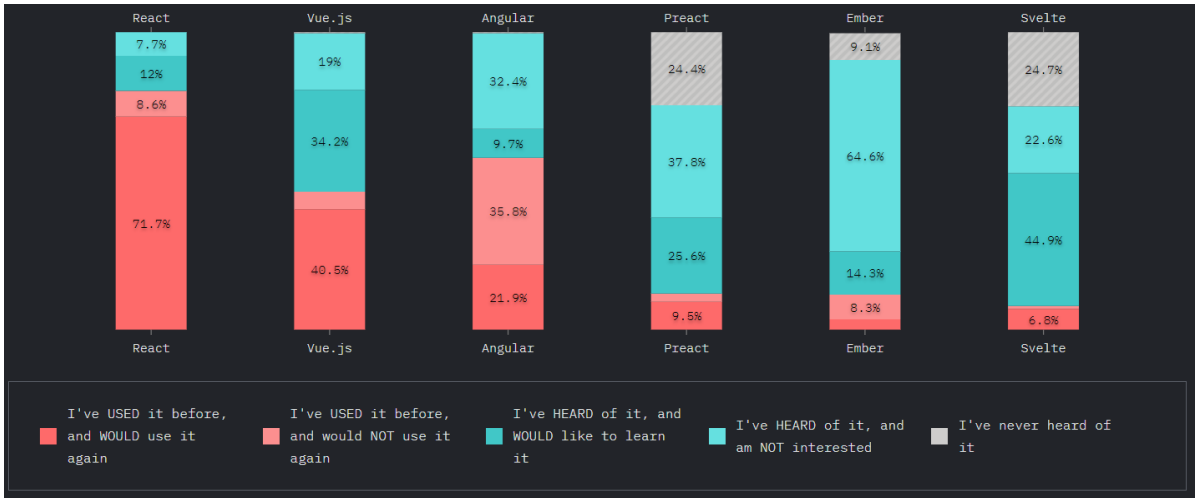

Figura 30: Opiniones sobre Front-end frameworks and libraries.
Tomado de [16]

Los segmentos más oscuros representan opiniones positivas, mientras que los segmentos más claros corresponden a opiniones negativas. Lo cual para más claridad se describe a continuación:

- Lo he usado antes, y lo volvería a usar.
- Lo he usado antes, y NO lo volvería a usar.

- Lo he oído, y me gustaría aprenderlo.
- Lo he oído y no me interesa.
- Nunca he oído hablar de ello.

5.1.4. Visual Studio Code

Visual Studio Code, cuyo logotipo se muestra en la Figura 31, es un editor de código fuente ligero pero potente que se ejecuta en su escritorio y está disponible para Windows, macOS y Linux. Viene con soporte incorporado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes (como C ++, C #, Java, Python, PHP, Go) y tiempos de ejecución (como .NET y Unity) [17].

Figura 31: Logotipo de Visual Studio Code

El editor de código de Visual Studio es compatible con React.js IntelliSense y la navegación de código lista para usar, además de contar con diversas extensiones o complementos para un mejor desarrollo de aplicaciones.

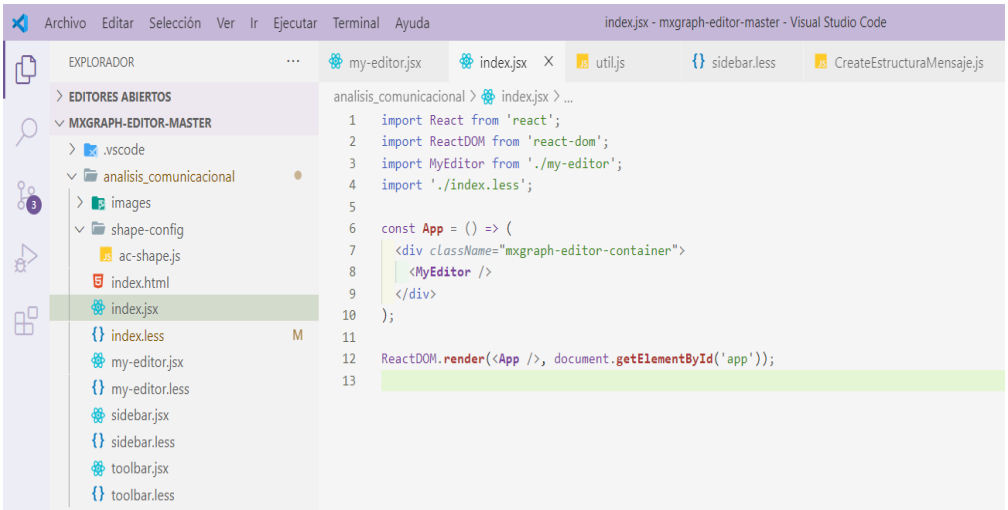

Figura 32: Visual Studio Code - Ejemplo

5.1.5. GitLab

GitLab, cuyo logotipo se muestra en la Figura 33, GitLab, es un repositorio de gestión de proyectos dotado de interfaz web. Como

podemos deducir del nombre, está construido sobre Git, y básicamente nos proporciona el código para generar un servidor y gestionar los clientes, sus opciones y los servicios ofrecidos.

Figura 33: Logotipo de GitLab

A través de GitLab, podemos gestionar grupos, personas y los permisos que queremos que tengan los usuarios dentro de los grupos o proyectos a los que pertenezcan. También nos permite llevar a cabo un seguimiento del estado actual y del histórico de los proyectos pudiendo, así, ver todos los cambios y modificaciones producidas en el tiempo de desarrollo, además de gráficos, otros datos de interés de los proyectos y servicios que van más allá del control de versiones. Ejemplos de estos servicios serían los comentarios de usuarios sobre un proyecto, herramientas de planificación, issues (utilizados para reportar o avisar de errores), requests (para facilitar a la comunidad de proyectos compartidos, se permite que la gente haga peticiones de actualización con su código y que, si al propietario del proyecto le parece adecuado, puedan aceptarse), etc. [18]

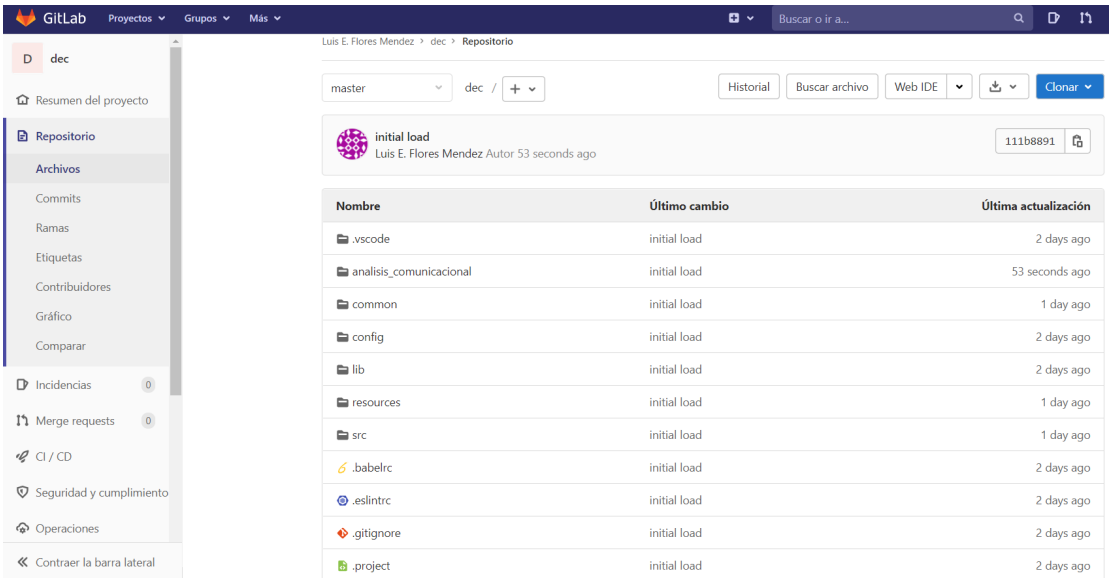

Figura 34: Repositorio del proyecto – GitLab

Capítulo 6

Validación

En este capítulo nos centraremos en la validación de la herramienta, con la finalidad de validar que los objetivos se han cumplido. Para ello se ha utilizado un diagrama de ejemplo para la su modelación en ambas herramientas tanto la propuesta como en la existente y comprobar a través de un cuestionario si la herramienta propuesta es una mejora a la herramienta existente.

6.1. Diseño de la Validación

6.1.1. Objetivo de la validación

Demostrar que la herramienta propuesta es una mejora de la herramienta existente, satisfaciendo las necesidades de los usuarios a través de las pruebas y la usabilidad.

6.1.2. Pregunta de investigación

¿Se encuentran los usuarios más satisfechos con la herramienta propuesta respecto a la herramienta existente?

6.1.3. Hipótesis

Para responder a la pregunta de investigación se plantea las siguientes hipótesis:

- H_0 = No Existe satisfacción con la herramienta propuesta.
- H_1 = Existe satisfacción con la herramienta propuesta.

6.1.4. Variables

- PEU: Facilidad de uso percibida
- PU: Utilidad percibida
- IU: Intención de uso

6.1.5. Instrumento

En [19] se propone un modelo teórico y un instrumento de medición para evaluar los métodos de diseño de sistemas de información; por el cual teniendo en cuenta ello, para dar respuesta a la hipótesis se realizó la evaluación de la herramienta propuesta a través de un cuestionario

como instrumento de investigación que consiste en nueve (09) preguntas ocho (08) con escalas del 1 al 5 (1=muy en desacuerdo y 5=muy de acuerdo) y una (01) pregunta de manera general por el usuario para conocer la opinión sobre la herramienta propuesta, con la finalidad de recopilar información para el análisis correspondiente y conocer si los usuarios adoptarían por la herramienta propuesta. Las preguntas del cuestionario han sido las siguientes:

1. Encuentro que la herramienta para aplicar el modelado de procesos de negocio con análisis comunicacional no es compleja y es fácil de usar.

2. Definitivamente usaría esta herramienta para modelar procesos de negocio con análisis comunicacional.

3. Me siento cómodo utilizando esta herramienta.

4. La interfaz de la herramienta es agradable.

5. La herramienta cuenta con las funciones necesarias para el modelado de negocio con análisis comunicacional.

6. De manera general estoy satisfecho con la herramienta.

7. Fue fácil aprender a usar esta herramienta.

8. En General encuentro que la herramienta es útil.

9. De manera general cuál es su opinión sobre la herramienta (Escriba su Comentario).

Para cada pregunta existe un comentario.

6.1.6. Procedimiento

Dos (02) usuarios expertos con conocimientos en Análisis Comunicacional, participaron como validadores, y se encargaron de validar la herramienta implementada de modelado de Análisis Comunicacional basada en Web frente a la herramienta GREAT Process Modeller el cual se encuentra dentro de una máquina virtual con el Sistema Operativo XP.

A pesar de ser conscientes de que dos evaluadores puede ser un número muy bajo, no ha sido fácil encontrar más participantes del perfil deseado en el ámbito del proyecto, y estos evaluadores proporcionan en cualquier caso un punto de partida interesante, marcando la pauta de lo que podría ser una validación más extensa y rigurosa.

Los usuarios se encargaron de realizar un diagrama de ejemplo (ver Figura 35) dado y teniendo en cuenta la estructura de mensaje en ambas herramientas para luego puedan contestar el formulario de evaluación estructurado por un cuestionario para luego poder analizar los resultados. Para ello se usa un problema de tamaño pequeño pero que permite a los usuarios poder evaluar toda la herramienta.

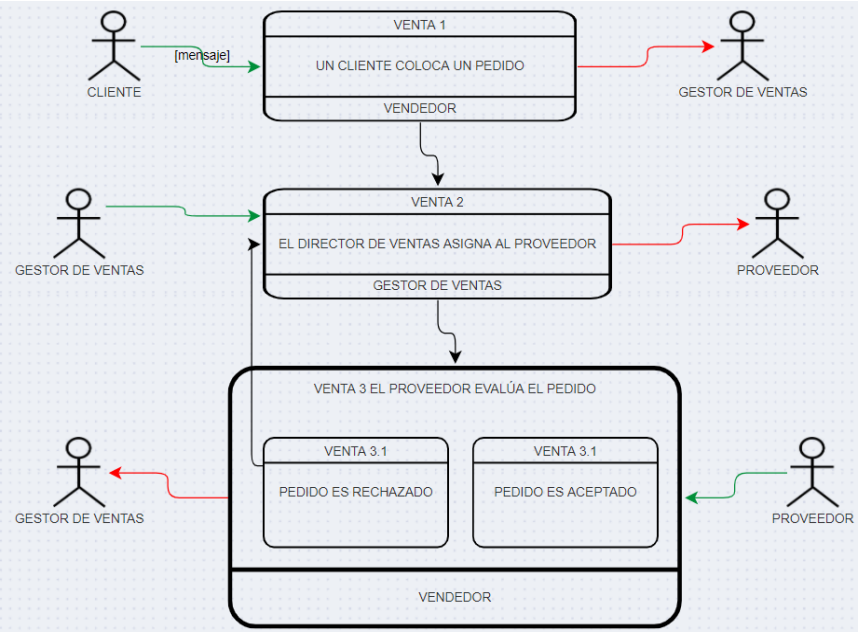

Figura 35: Diagrama de Ejemplo - Validación

6.2. Resultados

Los usuarios respondieron a las preguntas obteniéndose los siguientes resultados:

1. **Encuentro que la herramienta para aplicar el modelado de procesos de negocio con análisis comunicacional no es compleja y es fácil de usar.**

USUARIO/VALORACIÓN	1	2	3	4	5
Usuario 1					X
Usuario 2					X

2. **Definitivamente usaría esta herramienta para modelar procesos de negocio con análisis comunicacional.**

USUARIO/VALORACIÓN	1	2	3	4	5
Usuario 1			X		
Usuario 2			X		

3. **Me siento cómodo utilizando esta herramienta.**

USUARIO/VALORACIÓN	1	2	3	4	5
Usuario 1				X	
Usuario 2					X

4. **La interfaz de la herramienta es agradable.**

USUARIO/VALORACIÓN	1	2	3	4	5
Usuario 1					X
Usuario 2					X

5. **La herramienta cuenta con las funciones necesarias para el modelado de negocio con análisis comunicacional.**

USUARIO/VALORACIÓN	1	2	3	4	5
Usuario 1				X	
Usuario 2			X		

6. **De manera general estoy satisfecho con la herramienta.**

USUARIO/VALORACIÓN	1	2	3	4	5
Usuario 1			X		
Usuario 2			X		

7. **Fue fácil aprender a usar esta herramienta.**

USUARIO/VALORACIÓN	1	2	3	4	5
Usuario 1					X
Usuario 2					X

8. En General encuentro que la herramienta es útil.

USUARIO/VALORACIÓN	1	2	3	4	5
Usuario 1				X	
Usuario 2					X

9. De manera general cuál es su opinión sobre la herramienta (Escriba su Comentario).

a. El Usuario 1 comento lo siguiente: Creo que una de las principales ventajas de esta herramienta es que es fácil de comprender y la mayoría de sus funcionalidades se pueden hacer de forma intuitiva. Sin embargo, al compararla con la herramienta GREAT Process Modeller, falta incorporar algunos elementos para el diseño de diagramas de análisis comunicacionales y además solucionar algunos fallos en la implementación, como validación de los diagramas, la enumeración de actores/eventos y la forma poco intuitiva de moverse en el modelador.

   Por otro lado, GREAT permite una mayor fluidez al momento diagramar, ya que se pueden ir incorporando nuevos eventos comunicativos u otros elementos, a partir de un actor u evento.

b. El Usuario 2 comento lo siguiente: En comparación con la herramienta GREAT, faltan funcionalidades: OR-AND-START-END. Además, existe un problema con la declaración de estructuras de mensaje.

   Sólo se puede agregar una estructura de mensaje por diagrama. Si se elimina la estructura de mensaje creada, no permite agregar otra. Por otro lado, la herramienta advierte mensajes respecto a elementos eliminados del diagrama, y al seleccionar una opción del toolbar, se destacan todas. Finalmente, si creo un elemento fuera del panel, no me permite hacer scroll.

6.3. Discusión

Se procedió a valorar los resultados, identificando las preguntas en la cuales la valoración de los participantes es menor que el nivel satisfactorio (4) y las discrepancias de opinión.

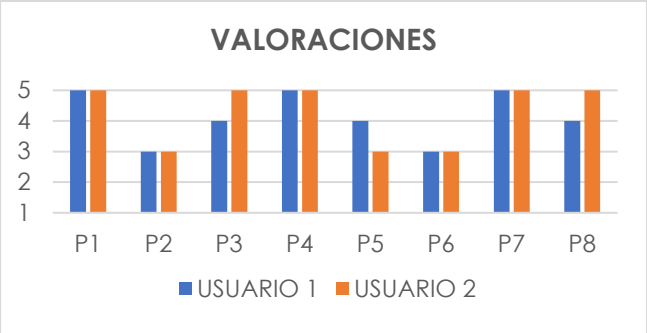

Figura 36: Valoraciones de la Validación

Al analizar las preguntas en la que ambos usuarios manifiestan una valoración igual o mayor a 4, es posible afirmar que existe una opinión positiva respecto a la usabilidad de la herramienta. Agrupando las valoraciones en los cuales los usuarios se manifiestan satisfechos con una valoración entre cuatro (4) o cinco (5); empezamos con la valoración en **P1** los usuarios coinciden que la herramienta no es compleja y es muy fácil de usar, en **P3** de manera general vemos que los usuarios al usar la herramienta se sienten cómodos, en **P4** los usuarios coinciden que la herramienta tiene una interfaz muy agradable, en **P7** los usuarios coinciden en que la herramienta es muy fácil de aprender.

Otro aspecto valorado positivamente es la intención de utilizar la herramienta para el modelado de análisis comunicacional, Esto se manifiesta en **P8** en que los usuarios encuentran útil la herramienta.

Respecto a las preguntas en las cuales no existe una valoración positiva (valoración igual o menos a tres), en **P6** vemos que a nivel general no hay una satisfacción absoluta con la herramienta, lo que puede estar relacionado con la respuesta similar a **P2**, que no hay una disposición de los usuarios para modelar procesos de negocio con análisis comunicacional con esta herramienta. Esto puede deberse a que los usuarios no se sienten cómodos con el método, mas que con la herramienta, y por ello en términos generales no se siente satisfechos. Otra interpretación es que los usuarios requieran funcionalidades no presentes en la herramienta que han sido descritas en los comentarios de la **P9**; como, por ejemplo: la falta de los elementos OR-AND-START-END, validación de los diagramas, la enumeración de actores/eventos y la forma poco intuitiva de moverse en el modelador.

En **P5**, podemos ver la valoración de algo de acuerdo y ni de acuerdo ni en desacuerdo con respecto a la herramienta si cuenta con las funciones necesarias para el modelado de negocio con análisis comunicacional debido a los comentarios realizados en **P9**; como, por ejemplo: la falta de los elementos OR-AND-START-END.

Teniendo en consideración lo mencionado anteriormente; en general, podemos decir que la herramienta no es compleja, es fácil de usar y

aprender, contando con una interfaz agradable para los usuarios, permitiéndoles un control fácil de las opciones que la herramienta presenta; a su vez, los usuarios se sienten cómodos y lo ven de utilidad la herramienta propuesta.

Sin embargo, la herramienta cuenta con valoraciones no esperadas debido a las observaciones mencionadas por los usuarios que validaron la herramienta en la pregunta número nueve (9); esto se debe a que existen tres definiciones de análisis comunicacional en [1] no se cuenta con dichos conectores, en [5] se cuenta con los conectores START, END y MERGE "OR" y en la herramienta GREAT cuenta con todos los conectores (OR-AND-START-END). Para el diseño de esta herramienta se tuvo en cuenta [1], según Figura 37.

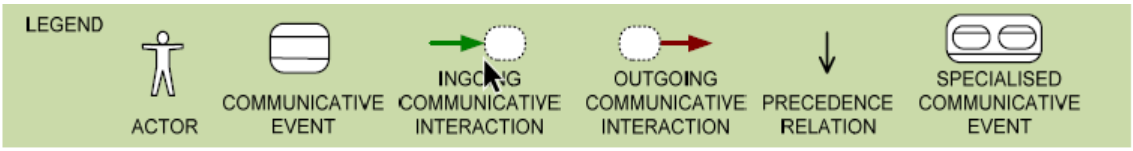

Figura 37: Leyenda de artículo usado como referencia [1]

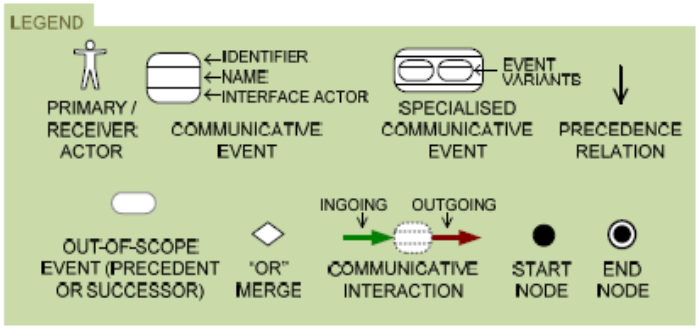

Figura 38: Leyenda en ejemplos en tesis que describe el método [2]

Figura 39: Leyenda de la Herramienta GREAT

Sin embargo; con el apoyo de los usuarios mediante sus opiniones se logró añadir a la herramienta propuesta dos elementos adicionales Inicio (Start) y Fin (End) para el modelado de procesos de negocio con Análisis Comunicacional, dejando para futuro la implementación de los conectores OR y AND.

Capítulo 7

Conclusiones y Trabajo futuro

Con el diseño e implementación de este proyecto se dispone de una herramienta de soporte para el modelado de procesos de negocio con Análisis Comunicacional considerando los diagramas de eventos comunicativos con las estructuras de mensajes.

Para el diseño e implementación de la herramienta se hizo uso de las librerías mxGraph y React y como IDE Visual Studio Code que fue de gran ayuda para para lograr esto.

La herramienta que se diseñó e implemento ofrece una interfaz agradable, no es compleja, es fácil de usar y aprender permitiendo a los usuarios un control fácil de las opciones que dispone la herramienta; siendo esta de utilidad para los usuarios.

En términos generales, la herramienta propuesta cumple con los objetivos propuestos; por lo que se puede decir que puliendo los detalles indicados en las observaciones por los usuarios que validaron la herramienta e incorporando nuevos elementos, mejoraría las valoraciones y se haría aún más usable.

Con este trabajo se ha dado un paso en el modelado de procesos de negocios con Análisis Comunicacional (CA) esperando la integración con otros modelos.

La herramienta desarrollada es una contribución tanto para el desarrollo de un método holístico de producción de software, gracias a la capacidad de interoperabilidad de las tecnologías usadas, como a la masificación y estudio del método de Análisis Comunicacional.

Bibliografía

[1] España, Sergio & Gonzalez, Arturo & Pastor, Oscar. (2009). Communication Analysis: A Requirements Engineering Method for Information Systems. 530-545. 10.1007/978-3-642-02144-2_41.

[2] Noel, René & Panach, Ignacio & Pastor, Oscar. (2021). A Models-to-Program Information Systems Engineering Method. 10.1007/978-3-030-72696-6_8.

[3] R. J. Wieringa, Design science methodology for information systems and software engineering. Springer, 2014.

[4] España, Sergio & Gonzalez, Arturo & Pastor, Oscar & Ruiz, Marcela. (2011). A practical guide to Message Structures: a modelling technique for information systems analysis and design.

[5] S. España, Methodological integration of Communication Analysis into a model-driven software development framework. Tesis Doctoral, Universidad Politecnica de Valencia, Departamento de Sistemas Informáticos y Computación, 2011.

[6] GREAT Process Modeller. http://www.pros.webs.upv.es/great/ [accedido en noviembre 2020].

[7] H. Xiang, (2020) "Diseño y desarrollo de un sistema de Transformación de modelos basado en Web" [Trabajo realizado para optar al Título de Ingeniero Civil en Informática]. Universidad de Valparaíso.

[8] Introducción a JSON. https://www.json.org/json-es.html [accedido en noviembre 2020].

[9] S. Aguirre, (2020) JSON - Vol.1: Primeros pasos - Sintaxis - Tipos de datos, RedUsers, ISBN 978-9874757937.

[10] Cris Kobryn & Grady Booch & Ivar Jacobson & Jim Rumbaugh. (2019). UML Distilled: A Brief Guide to the Standard Object Modeling Language, ISBN 0321193687, 9780321193681.

[11] M. Sanchez, (2012) JavaScript, Innovación Y Cualificación, ISBN 978-8495733184.

[12] F. Luna, (2019) JavaScript - Aprende a programar en el lenguaje de la web, RedUsers, ISBN 978-9874958082.

[13] D. Varella, Herramienta de Documentación de Infraestructuras Informáticas. Trabajo Final de Grado, Universidad Jaume I, 2017.

[14] A. Banks, (2017) Learning React: Functional Web Development with React and Redux, "O'Reilly Media, Inc.", ISBN 978-1491954577.

[15] React Tutorial: An Overview and Walkthrough. https://www.taniarascia.com/getting-started-with-react/ [accedido en noviembre 2020]

[16] Front End Frameworks. https://2019.stateofjs.com/front-end-frameworks/ [accedido en noviembre 2020]

[17] Visual Studio Code. https://code.visualstudio.com/docs [accedido en noviembre 2020]

[18] GitLab. https://inlab.fib.upc.edu/es/blog/descubriendo-gitlab [accedido en noviembre 2020]

[19] Moody, Daniel L., "The Method Evaluation Model: A Theoretical Model for Validating Information Systems Design Methods" (2003). ECIS 2003 Proceedings. 79.